

MESInterface IT

**mitsubishi**

Oracle MOC Extension Installation and User's Guide

The logo for the Mitsubishi Q series, featuring a large, stylized 'Q' followed by the word 'series' in a serif font. The logo is set against a background of overlapping gray squares, one of which has a textured pattern.

Mitsubishi  
Programmable Controller

**MELSEC-Q**

***MESInterface IT***  
***(VN-SWMIT1-E)***

## □ SAFETY PRECAUTIONS □

(Always read these instructions before using this equipment.)

Before using this product, please read this manual and the relevant manuals introduced in this manual carefully and pay full attention to safety to handle the product correctly.

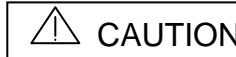
The instructions given in this manual are concerned with this product. For the safety instructions of the programmable controller system, please read the CPU module user's manual.

In this manual, the safety instructions are ranked as "DANGER" and "CAUTION".



**DANGER**

Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.



**CAUTION**

Indicates that incorrect handling may cause hazardous conditions, resulting in medium or slight personal injury or physical damage.

Note that the ! CAUTION level may lead to a serious consequence according to the circumstances.

Always follow the instructions of both levels because they are important to personal safety.

Please save this manual to make it accessible when required and always forward it to the end user.

### [Design Instructions]

#### **DANGER**

- When performing data changes or status control from the personal computer to the running PLC, configure up an interlock circuit outside the PLC system to ensure that the whole system will operate safely. In addition, predetermine corrective actions for the system so that you can take measures against any communication error caused by a cable connection fault or the like in online operations performed from the peripheral device to the PLC.

#### **CAUTION**

- Read the manual carefully before performing the online operations (especially forced output and operating status change) which will be executed with the personal computer connected to the running CPU module. Not doing so can damage the machine or cause an accident due to misoperation.

## REVISIONS

\* The manual number is given on the bottom left of the back cover.

| Print Date | * Manual Number | Revisions  |
|------------|-----------------|--|
| Feb., 2010 | BAD-804Q015-A0  | First edition  |
| Jun, 2012  | BAD-804Q015-A1  | <b>Correction</b><br>Update the screen shot of the Workbench |

This manual confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

This book applies to the Mitsubishi Electric Corporation. MESInterface IT product components and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Mitsubishi Electric Corporation. PROVIDES THIS BOOK "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions therefore, this statement may not apply to you.

This book could contain technical inaccuracies or typographical errors. Changes are made periodically to the information herein. Mitsubishi Electric Corporation. May make improvements and changes at any time to the product(s) and/or program(s) described in this book.

When you send information to Mitsubishi Electric Corporation., you grant Mitsubishi Electric Corporation. a nonexclusive right to use or distribute the information in any way it believes appropriate, without incurring any obligation to you.

## Table of Contents

|  |    |
|--|----|
| Preface.....   | ii |
| Who should read this book.....                           | ii |
| How this book is organized .....                         | ii |
| Related documentation .....                              | ii |
| Chapter 1: Introduction .....                            | 3  |
| Assumptions.....   | 3  |
| Chapter 2: Installing the Oracle MOC Extension .....     | 4  |
| Installing the Oracle MOC Package .....                  | 4  |
| Installing the Local Database Table .....                | 7  |
| Installing the Project .....                             | 9  |
| Chapter 3: Configuring for Your Oracle Environment ..... | 13 |
| Configure the Oracle Transport.....                      | 13 |
| Configure the Transport Map .....                        | 15 |
| Chapter 4: Customizing Transactions.....                 | 18 |
| Overview of the Projects .....                           | 18 |
| Oracle MOC Extension Trigger Structure.....              | 19 |
| Transaction ID .....                                     | 19 |
| Special Functions .....                                  | 19 |
| Data Collection .....                                    | 20 |
| Set Insert State .....                                   | 20 |
| Insert into Oracle .....                                 | 20 |
| Setting up a Direct Insert Transaction .....             | 20 |
| Setting the Transaction ID .....                         | 20 |
| Setting the Group ID .....                               | 21 |
| Adding Data Collection Actions .....                     | 22 |
| Setting up a Batch Insert Transaction .....              | 23 |
| Setting the Transaction ID .....                         | 24 |
| Setting the Group ID .....                               | 24 |
| Using a Counter Variable.....                            | 25 |
| Adding Data Collection Actions .....                     | 27 |
| Customizing the Subtriggers .....                        | 28 |
| Chapter 5: Troubleshooting .....                         | 29 |

# Preface

The Oracle MOC Extension is a MESInterface IT software configuration that allows communications with Oracle Manufacturing Operations Center.

This book describes how to install, configure and customize the Oracle MOC Extension in conjunction with the MESInterface IT product.

## ***Who should read this book***

This book is intended for engineers and programmers who will be defining and implementing data collection requirements to Oracle Manufacturing Operations Center using MESInterface IT. This book is also intended for people who manage device interface systems.

It is assumed that the reader has some experience with the MESInterface IT product.

## ***How this book is organized***

This book contains the following chapters:

*“Chapter 1: Introduction”* provides an overview of the Oracle MOC Extension and the assumptions for the reader.

*“Chapter 2: Installing the Oracle MOC Extension”* describes the installation process for the Oracle MOC Extension.

*“Chapter 3: Configuring for Your Oracle Environment”* describes the required configuration to communicate with your Oracle installation.

*“Chapter 4: Customizing Transactions”* describes how to customize the sample transactions into data collection transactions that meet your requirements.

*“Chapter 5: Troubleshooting”*

## ***Related documentation***

This book assumes you have access to the following related documentation:

- *MESInterface IT User's Guide*
- *MESInterface IT Enterprise User's Guide*
- *MESInterface IT System Administration User's Guide*
- *MESInterface IT Trigger Action Guide and Reference*
- *MESInterface IT Troubleshooting Guide*
- *MESInterface IT Quick Start - TCP*
- *MESInterface IT Quick Start – Database Insert*
- *MESInterface IT Quick Start – WebSphere MQ*

# Chapter 1: Introduction

Welcome to the Oracle Manufacturing Operations Center (MOC) Extension for MESInterface IT. The Oracle MOC Extension provides the components to collect and format data from plant floor devices and inserts it directly into Oracle Manufacturing Operation Center for processing. The Oracle MOC Extension has three components:

- **The Oracle MOC project import file** – contains the predefined transport, transport maps, and triggers to send data from a plant floor device to Oracle MOC.
- **The Oracle MOC local database table import file** – contains the predefined local database table for collecting data records prior to sending them to Oracle MOC.
- **The Oracle MOC package file** – contains special functions specifically required for the Oracle MOC interface.

This book provides step-by-step instructions on how to install and configure the Oracle MOC Extension components, along with suggestions on how to customize your Oracle MOC Extension installation to meet your specific data collection requirements.

## *Assumptions*

Before using the information in this book, the following should have occurred:

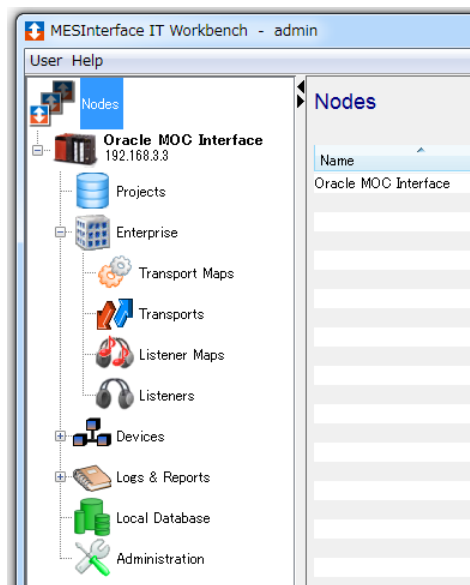
- A target MESInterface IT node has been installed, and is accessible on your network.
- The MESInterface IT Workbench has been installed and can communicate to the target MESInterface IT node.
- The Oracle Transport license has been installed on the target MESInterface IT node.
- The target MESInterface IT node has been licensed and configured to communicate with the plant floor devices identified for Oracle MOC data collection.
- The user is familiar with MESInterface IT terms, concepts, and Workbench operation.

# Chapter 2: Installing the Oracle MOC Extension

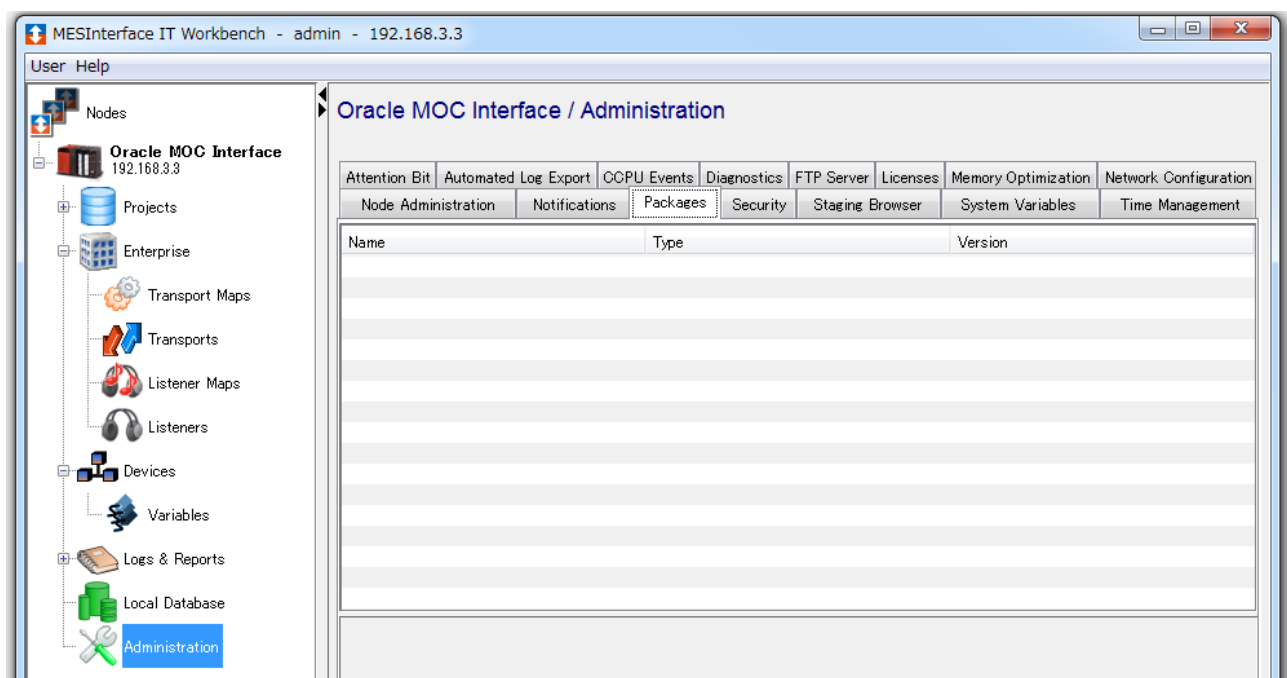
There are three components to the Oracle MOC Extension that need to be installed in your MESInterface IT node: a project, a local database table, and the Oracle MOC package.

## *Installing the Oracle MOC Package*

The first step is to install the Oracle MOC Package file. To start, open the MESInterface IT Workbench and select the node that you want to load the Oracle MOC Extension on.

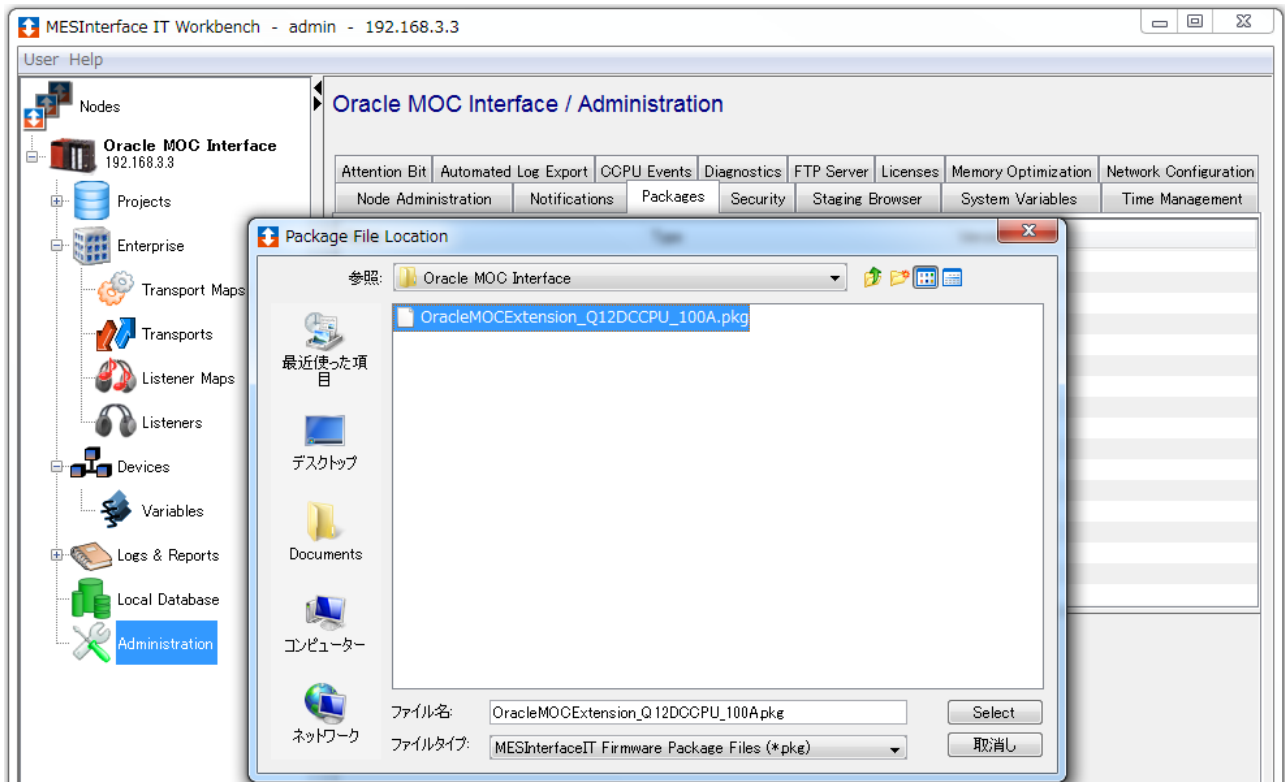


Then click on the Administration icon, and select the Packages tab.

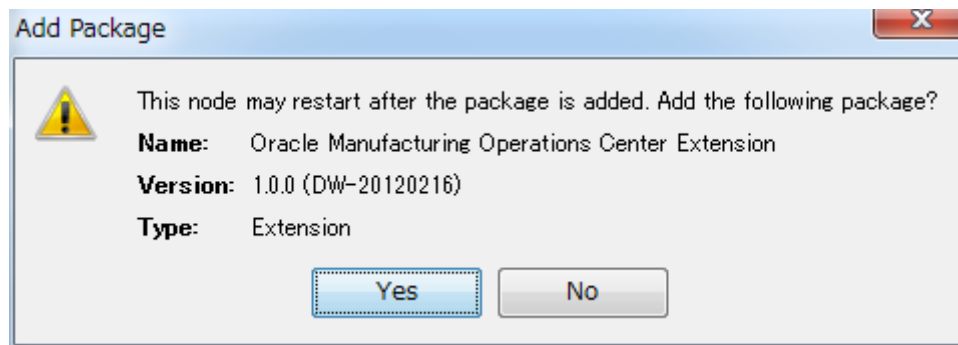




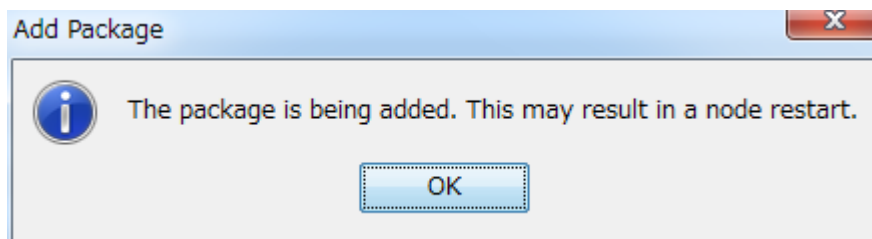
This will display your currently installed packages, and it may be empty. Click on the **Add** button and navigate to the Oracle MOC Extension folder. Select the **OracleMOCExtension\_(CPU name)\_(Version).pkg** file and click the **Select** button.



The following message box should appear. Click the Yes button.

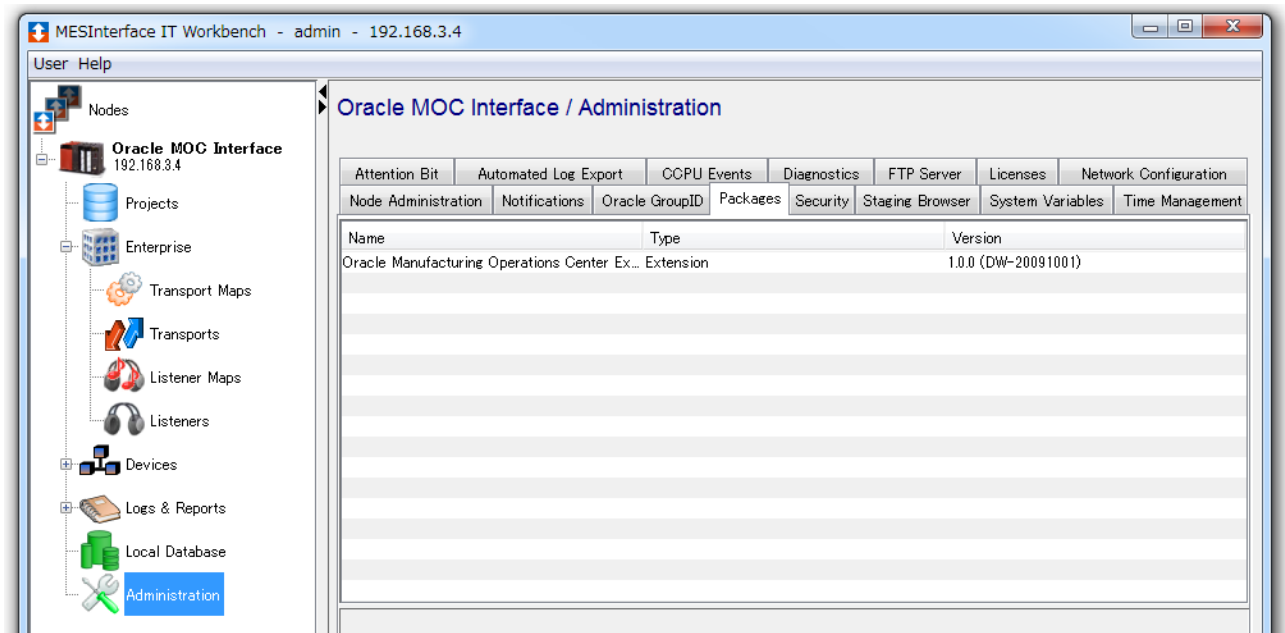


Once the installation is complete, the following message will appear. Click the Yes button and wait for the MESInterface IT node to restart. You may have to manually restart the MESInterface IT node.

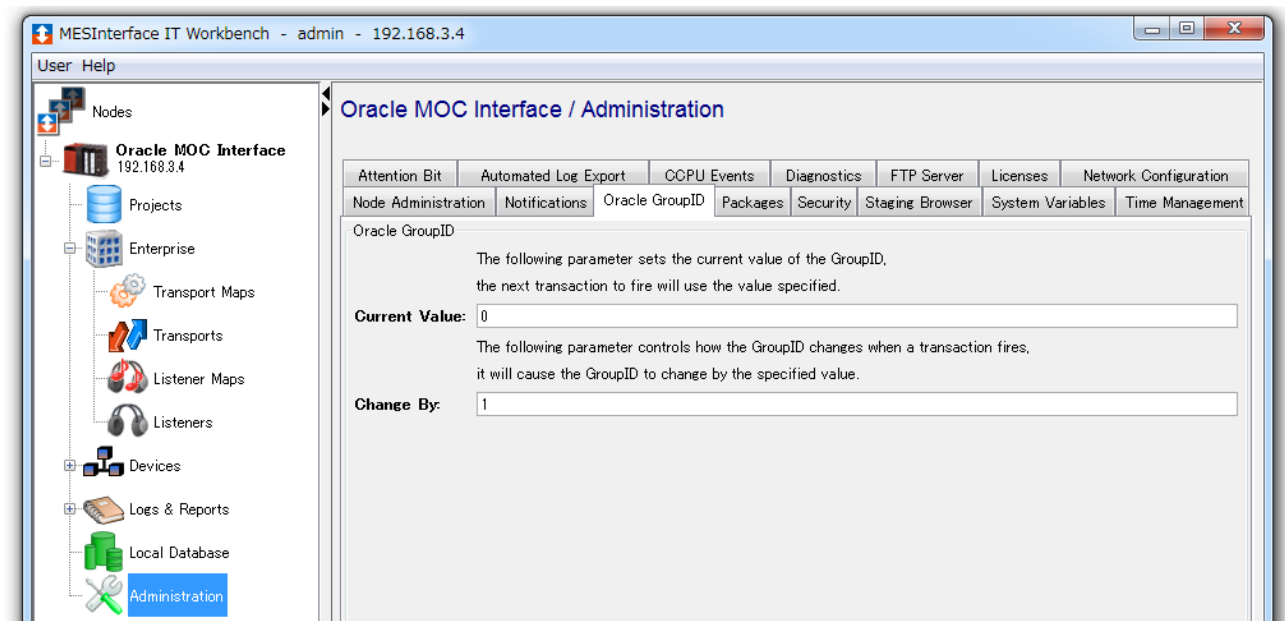


When the MESInterface IT node has restarted, return to the Administration panel, by clicking on the Administration icon.

Click on the **Packages** tab, and you should see an entry for the Oracle Manufacturing Operations Center package. You should also see a new tab, **Oracle GroupID**.



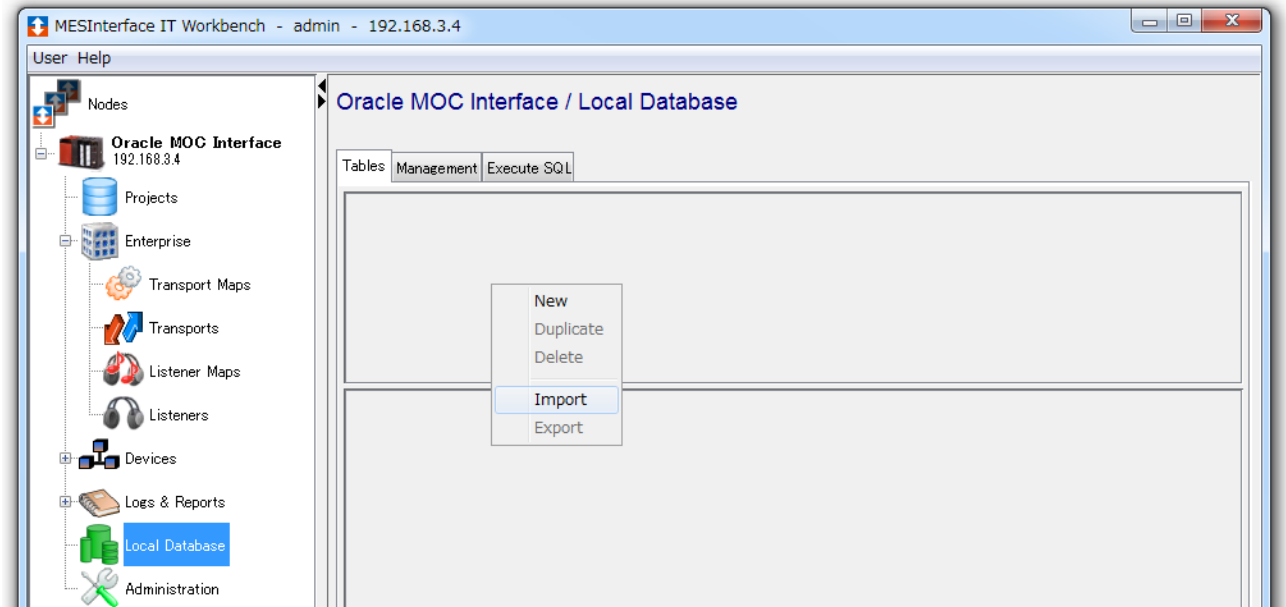
Click the **Oracle GroupID** tab and you should see the panel below. This panel will be used to customize your creation of the Oracle Group ID, and will be discussed in the **Customization** section of this document.



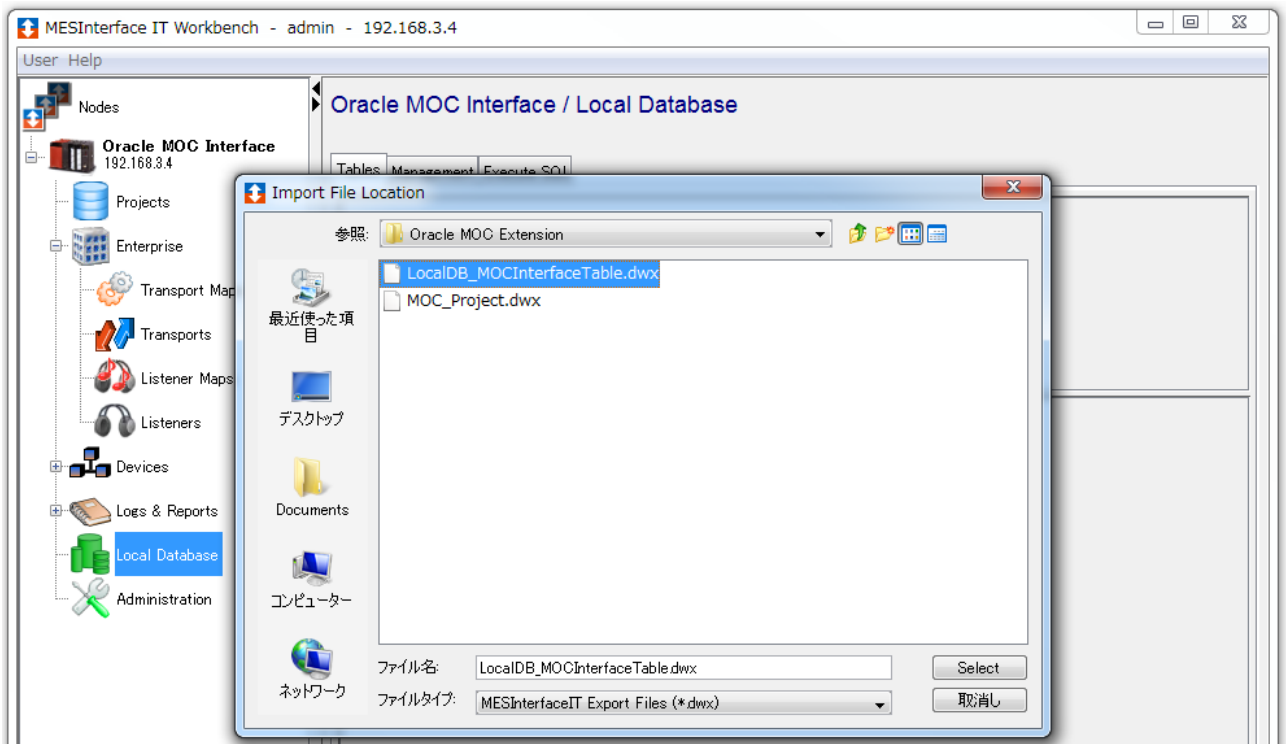
The local database table installation instruction will be discussed, next.

## Installing the Local Database Table

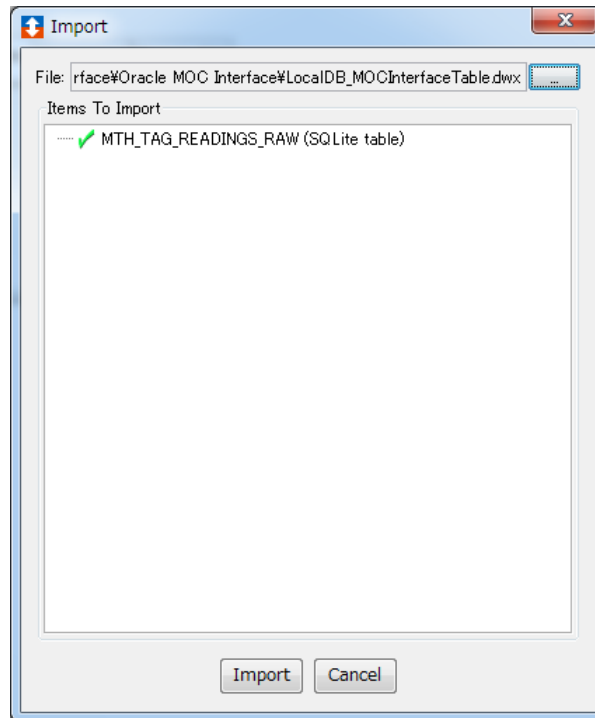
The second step of the Oracle MOC installation is to import a local database table that is used for batching data prior to sending it to the Oracle Manufacturing Operations Center interface table. Click on the **Local DB** icon. Place the mouse in the empty space and click the right mouse button. You should see a pop up selection list. Select **Import**.



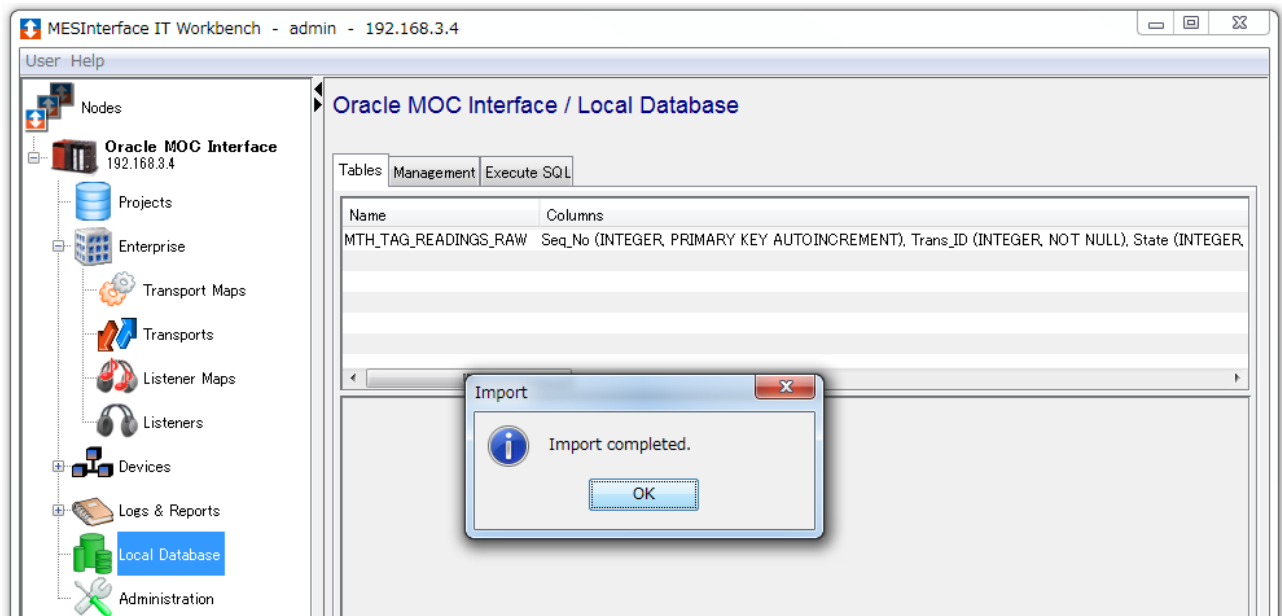
Navigate to the Oracle MOC Extension folder and select the **LocalDB\_MOCInterfaceTable.dwx** file and click the **Select** button.



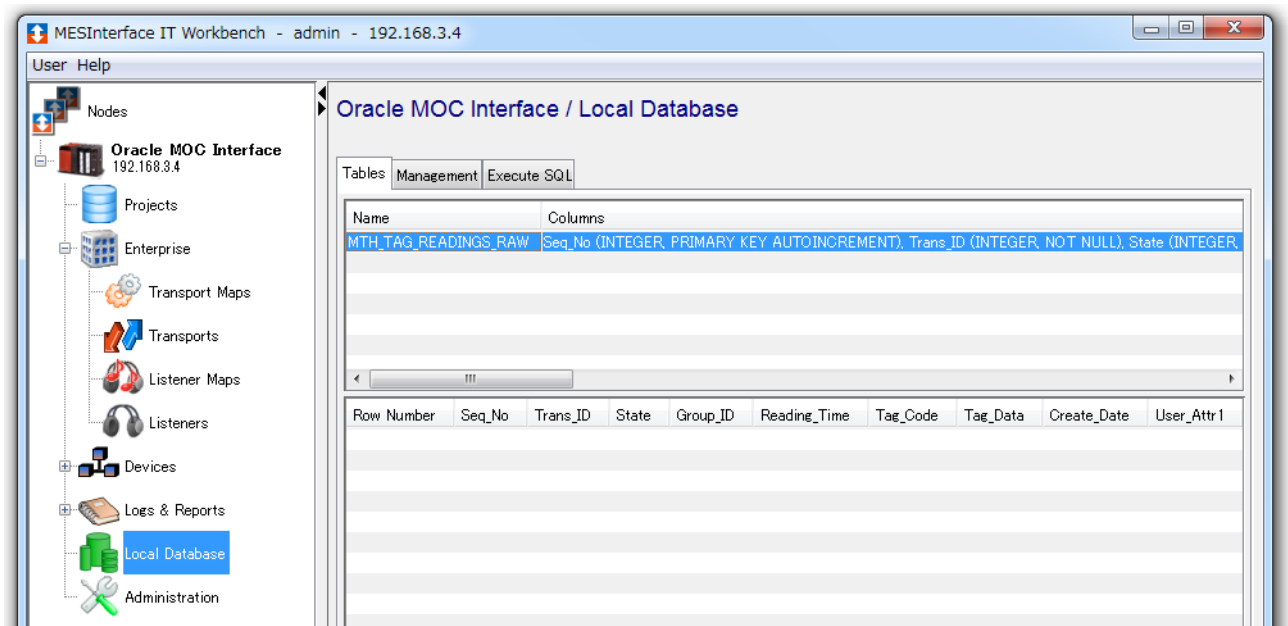
The following pop up should appear. Select the **Import** button.



When the import is finished, the **Import completed** message will be displayed. Click the **OK** button.



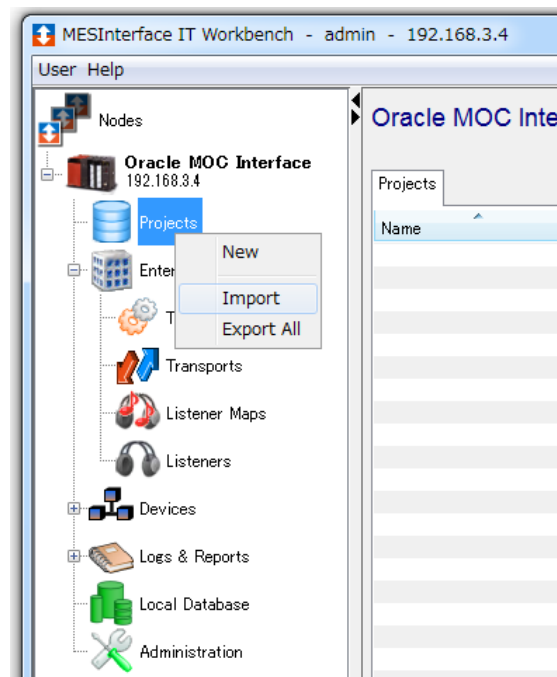
There should be a new table named MTH\_TAG READINGS\_RAW in the local database, as shown below.



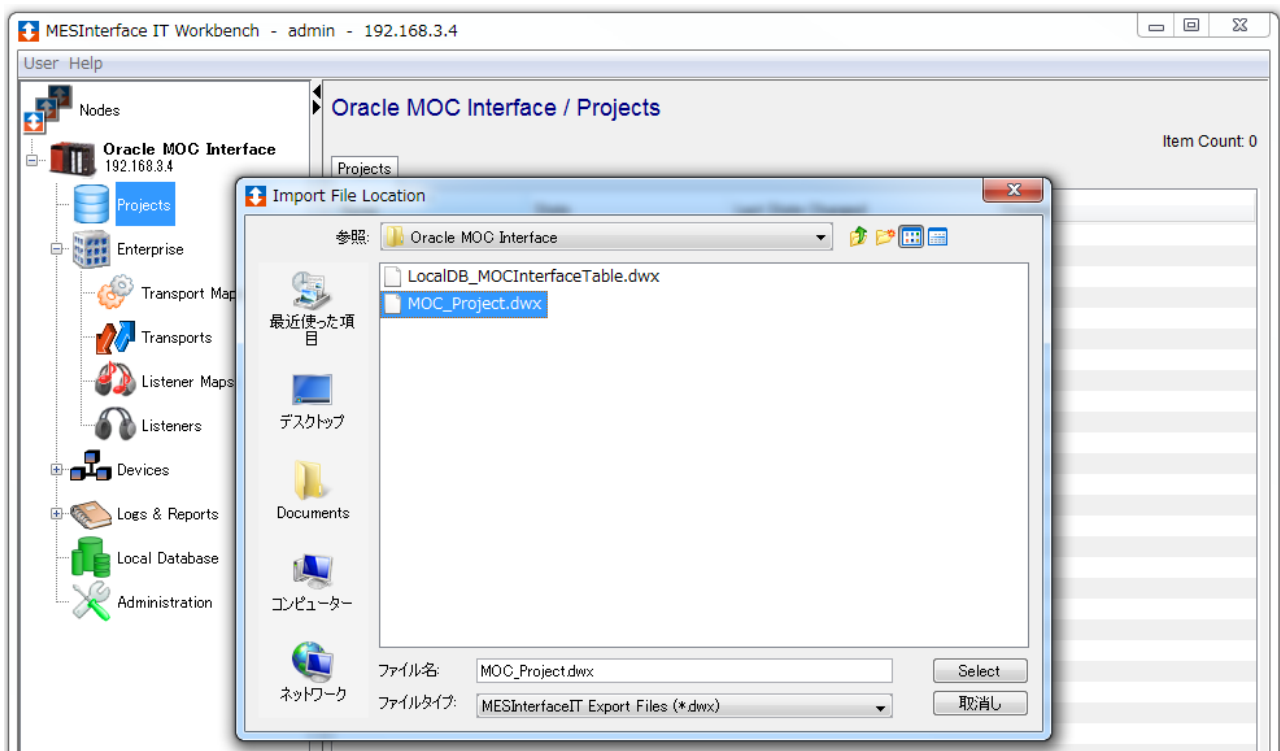
The local database table import is completed. Next, the project will be installed.

## *Installing the Project*

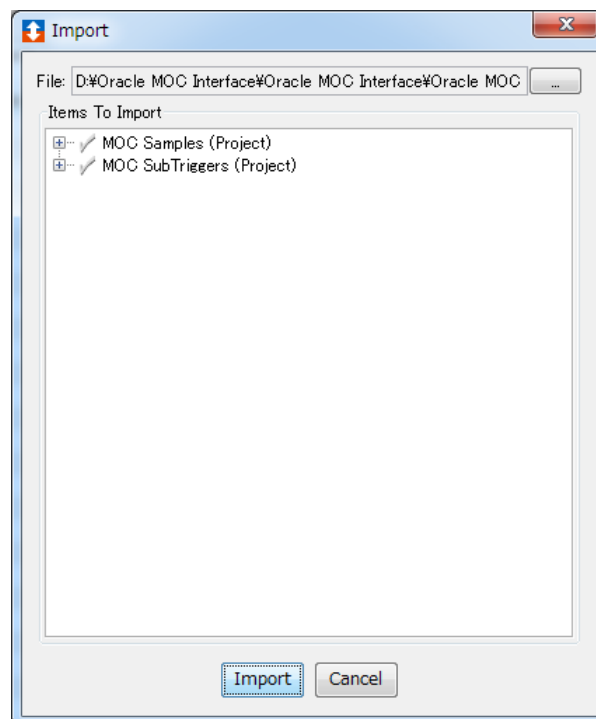
The last step of the installation imports the new project components for the Oracle MOC Extension. Start by placing the mouse over the Projects icon and press the right mouse button. Select **Import** from the list box.



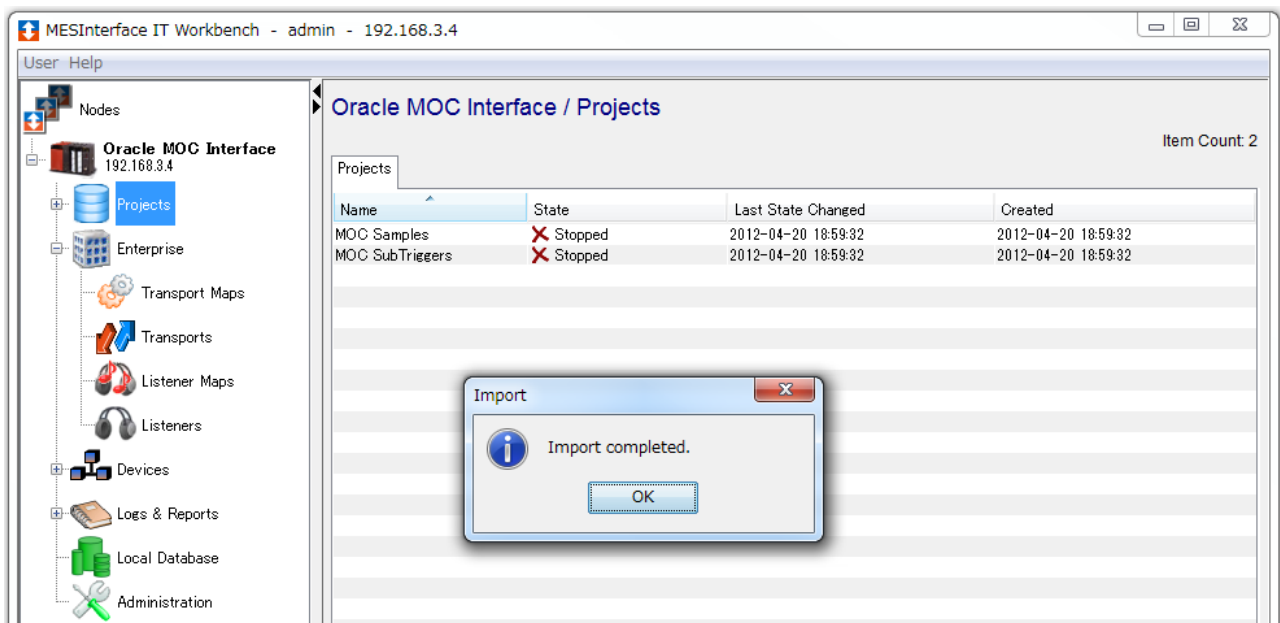
Navigate to the Oracle MOC Extension folder and select the **MOC\_Project.dwx** file and click the **Select** button.



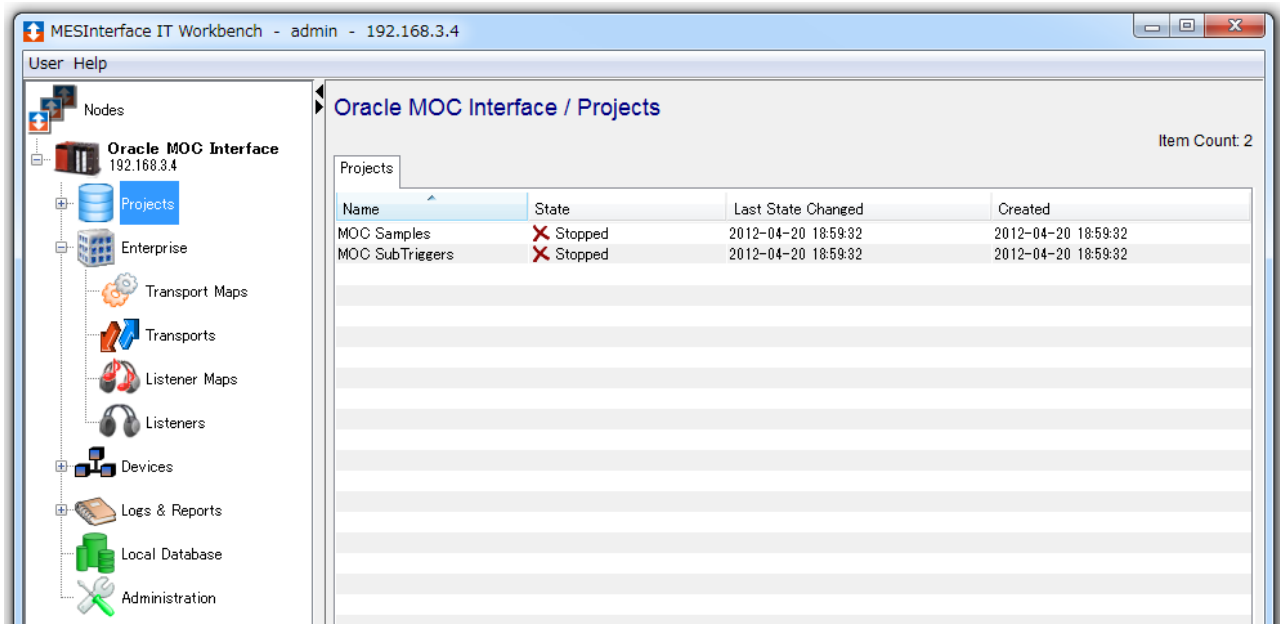
The following pop up should appear. Select the **Import** button.



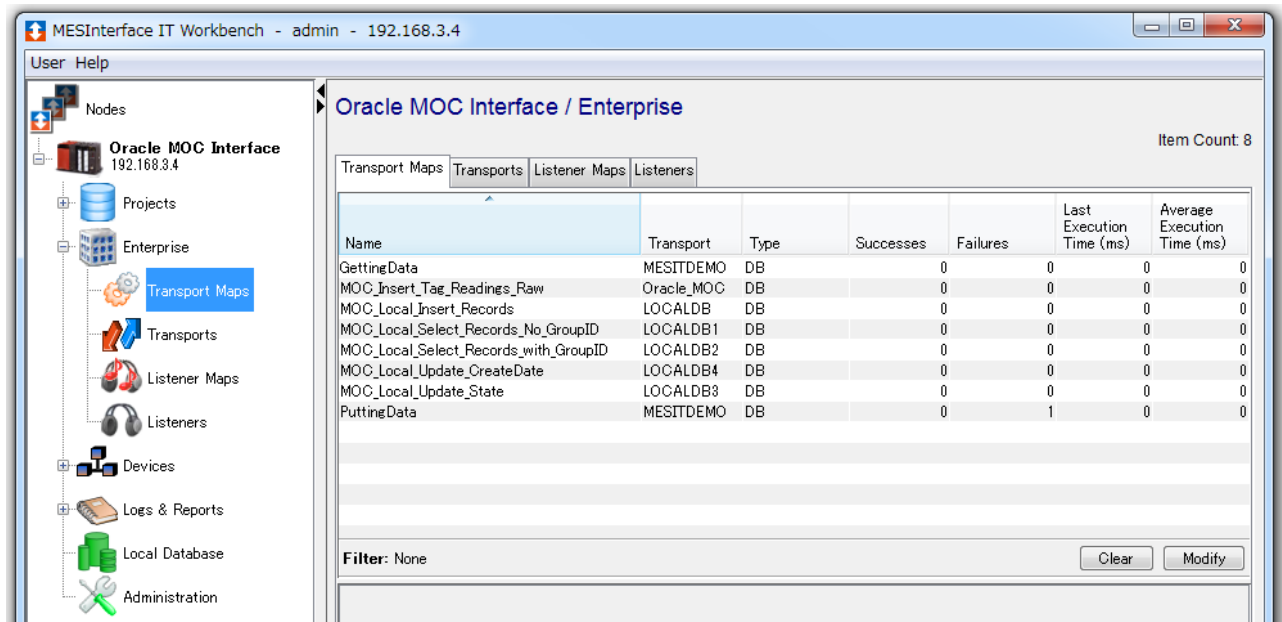
When the import is finished, the **Import completed** message will be displayed. Click the **OK** button.



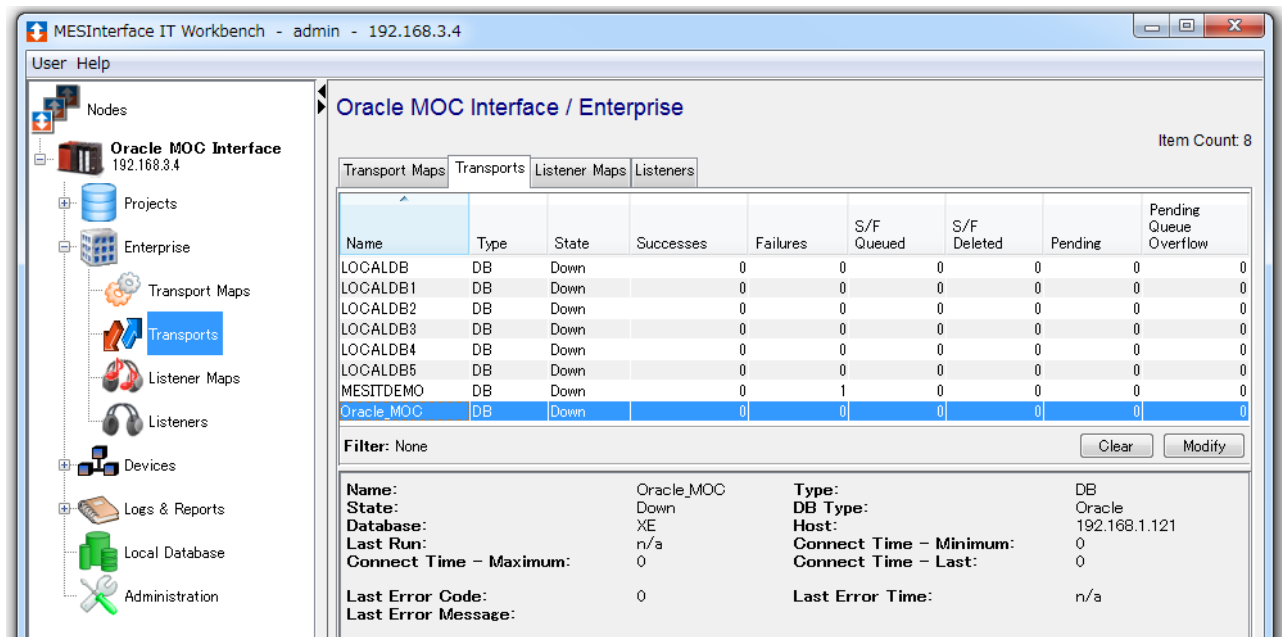
The import should create two projects, MOC Samples and MOC SubTriggers. Both of these projects will be discussed further in the Customization section of this document.



The import will also create six new transport maps, MOC\_Insert\_Tag\_Readings\_Raw, MOC\_Local\_Insert\_Records, MOC\_Local\_Select\_Records\_No\_GroupID, MOC\_Local\_Select\_Records\_with\_GroupID, MOC\_Local\_Update\_CreateDate, and MOC\_Local\_Update\_State. These will be discussed in the Configuration section of this document.



Finally, the import file will create a new transport, named Oracle\_MOC. We will discuss the configuring this transport for your environment in the next section.



This completes the installation of the Oracle MOC Extension.



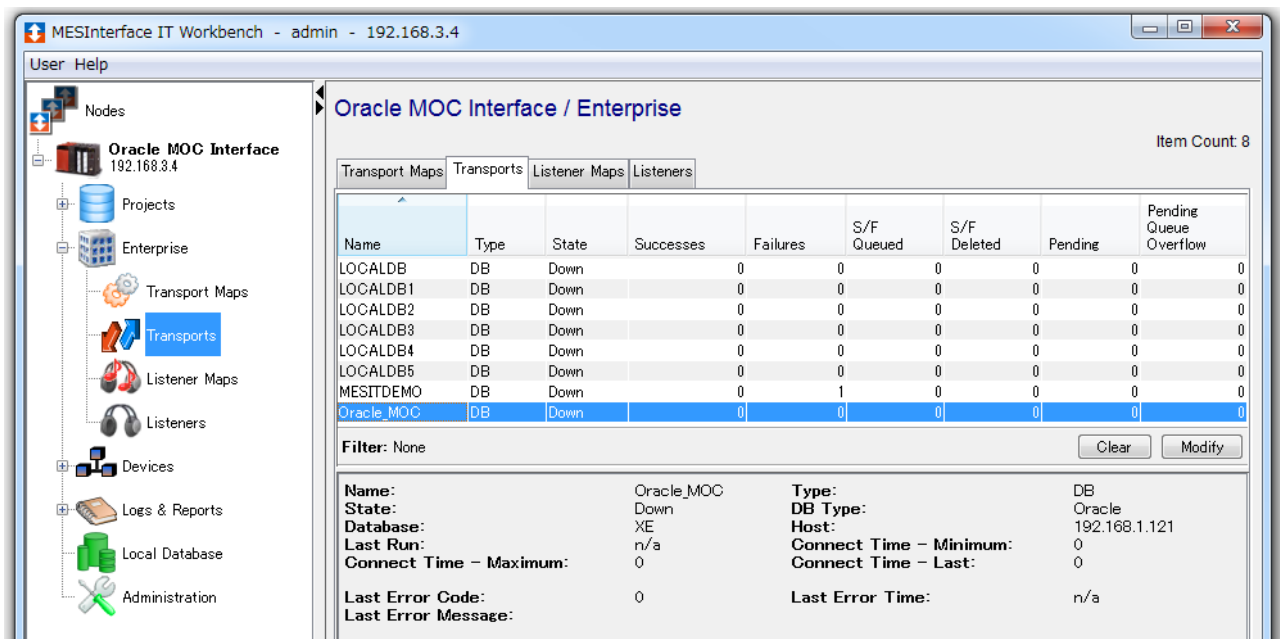
# Chapter 3: Configuring for Your Oracle Environment

There are two parts to configuring the Oracle MOC Extension. The first part is to modify the Oracle\_MOC transport to use your specific location and identification parameters. The second part is to modify the MOC\_Insert\_Tag\_Readings\_Raw transport map to use your instance of the Oracle Manufacturing Operations Center interface table. First we will configure the transport.

## Configure the Oracle Transport

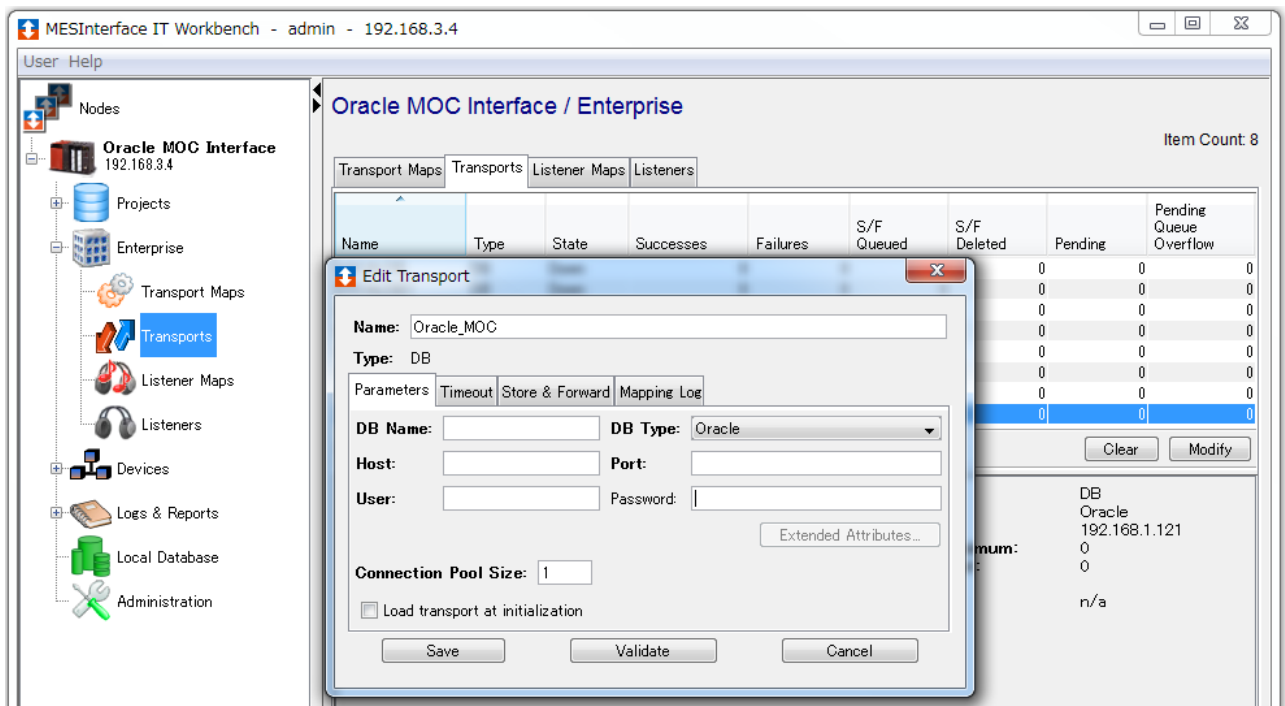
There are five key parameters which must be changed in the transport configuration to communicate with the Oracle Manufacturing Operations Center. These parameters are the database name, the hostname or IP address of the server, the port ID for the Oracle database, the user name, and the password to allow access to the database. Once these parameters are changed and validated, the database communication will be established.

The first step in configuring the transport is to click on the **Transports** icon in the left column, and select the **Oracle\_MOC** transport so it is highlighted, as shown below.

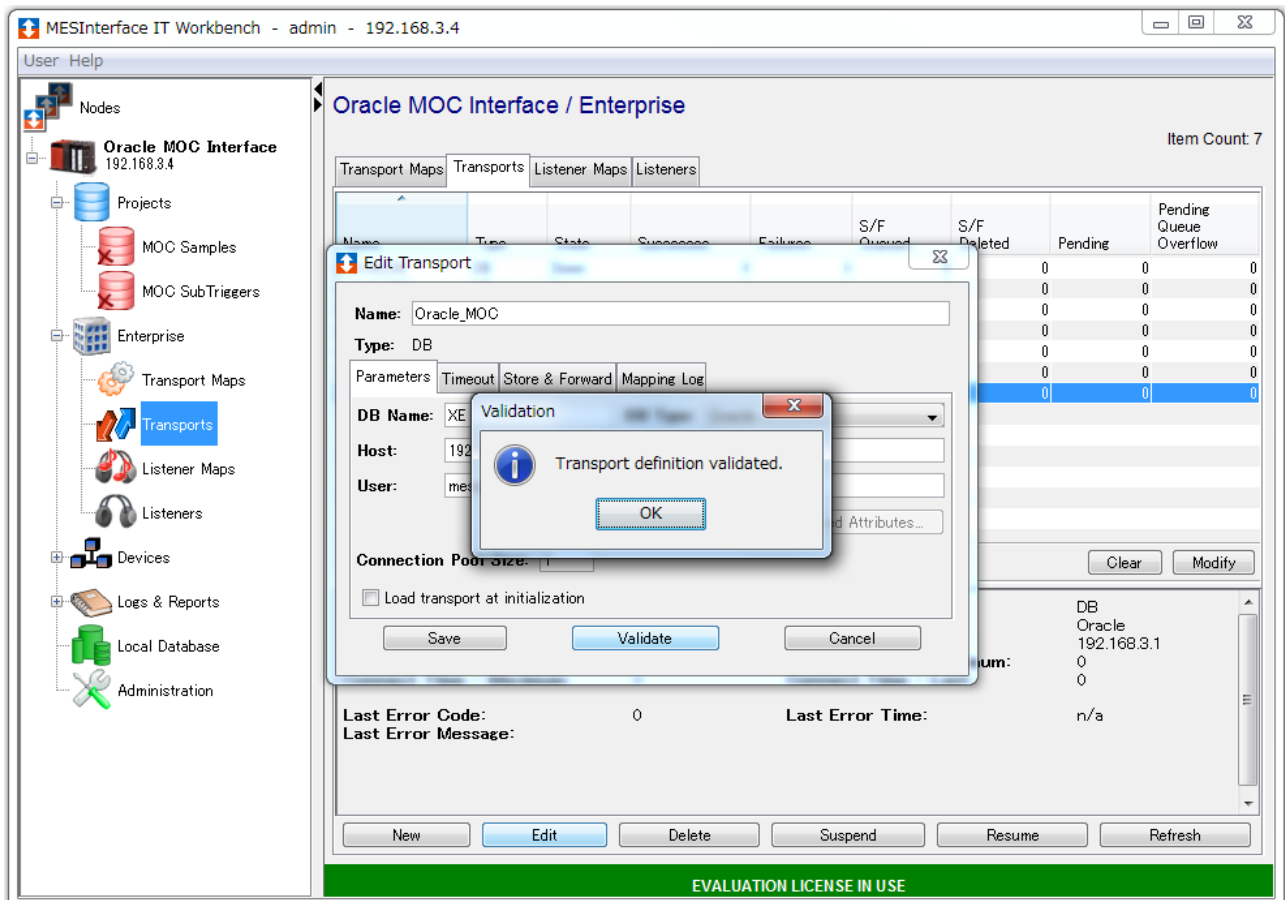


Then click the **Edit** button, and the edit window will be displayed.

Clear the DB Name, Host, Port, User ID, and Password fields and enter the specific information for your Oracle Manufacturing Operations Center installation.

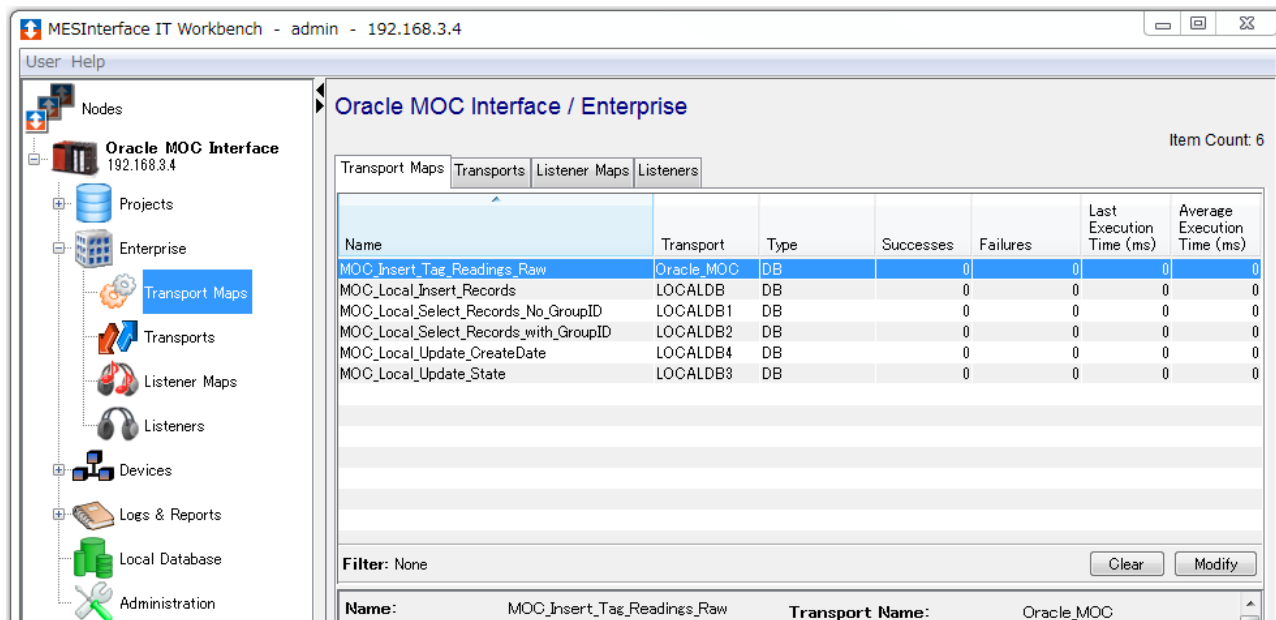


After you have entered the information, click the **Validate** button. If the MESInterface IT node is able to connect, you will see the message below. If it fails, please check the parameters and try again.

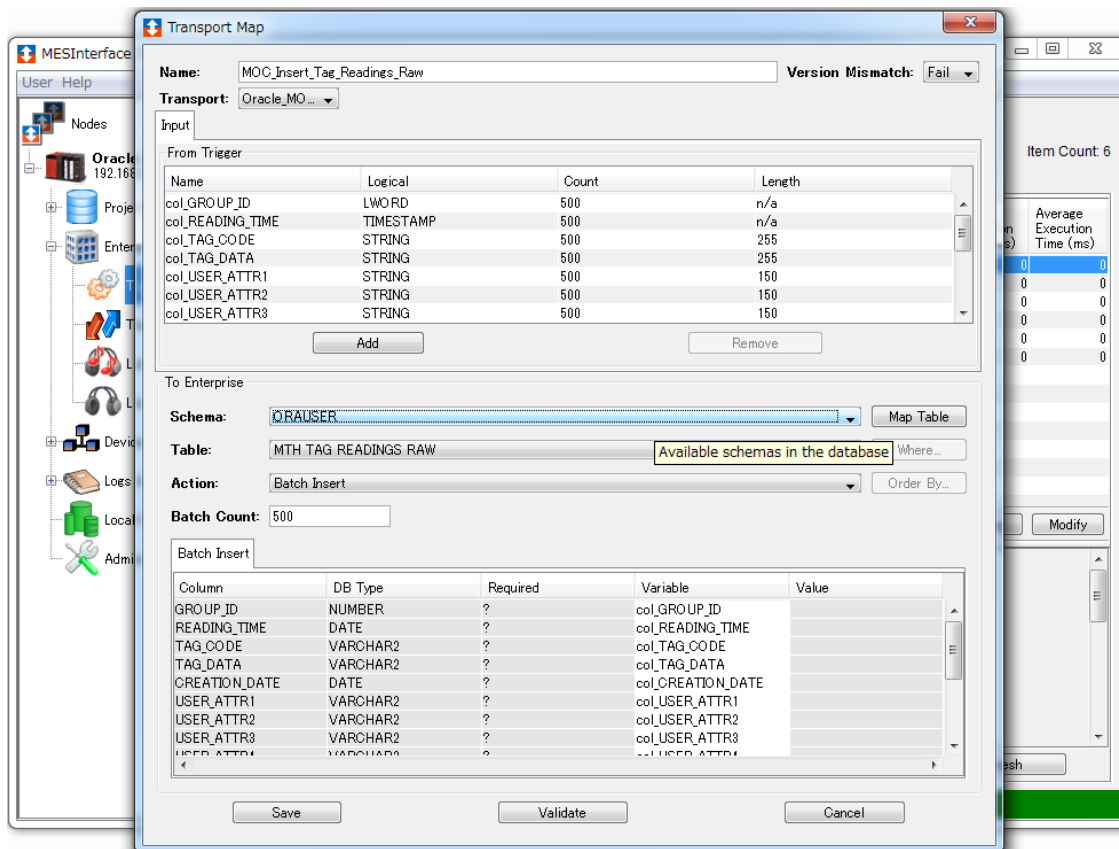


## Configure the Transport Map

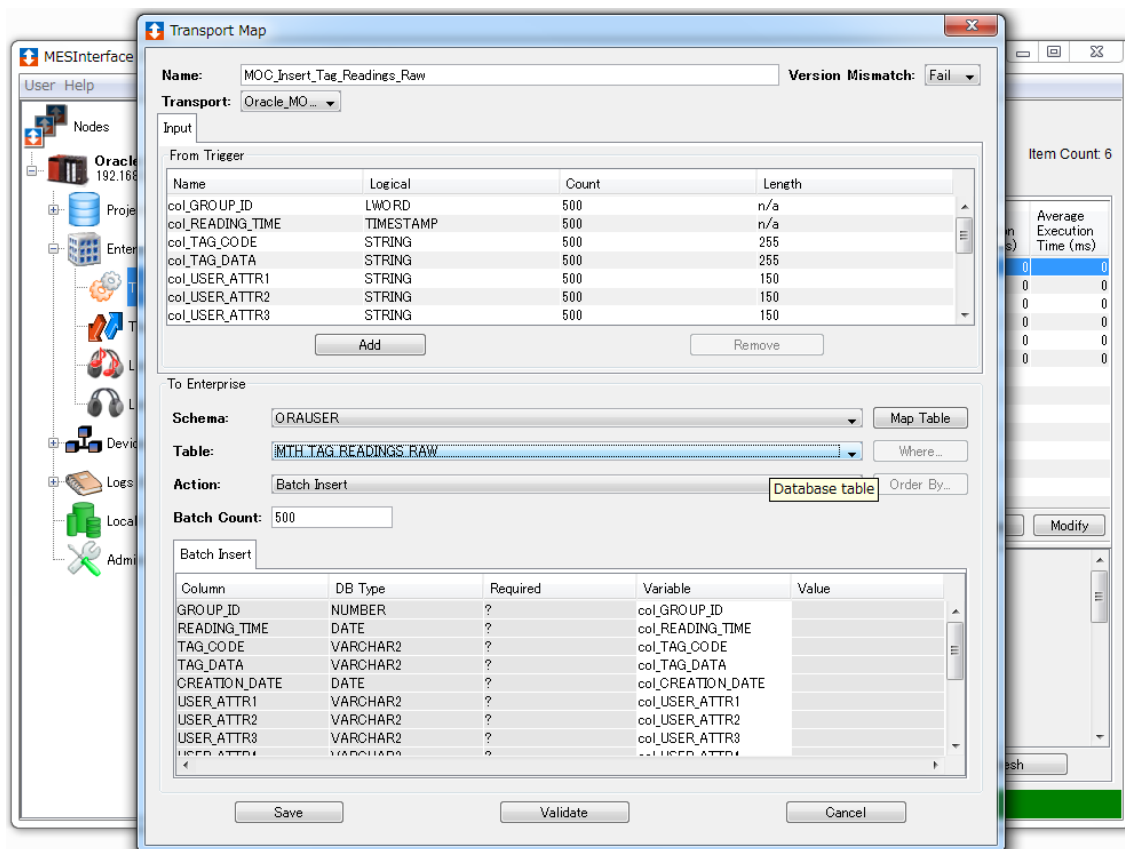
There is only one transport map, the **MOC\_Insert\_Tag\_Readings\_Raw** that needs to be configured. The modifications that need to be made are to select the schema where the Oracle Manufacturing Operations Center interface table is located, select the table and map the table. To begin, click the **Transport Maps** icon, select the **MOC\_Insert\_Tag\_Readings\_Raw** transport map, and click the **Edit** button.



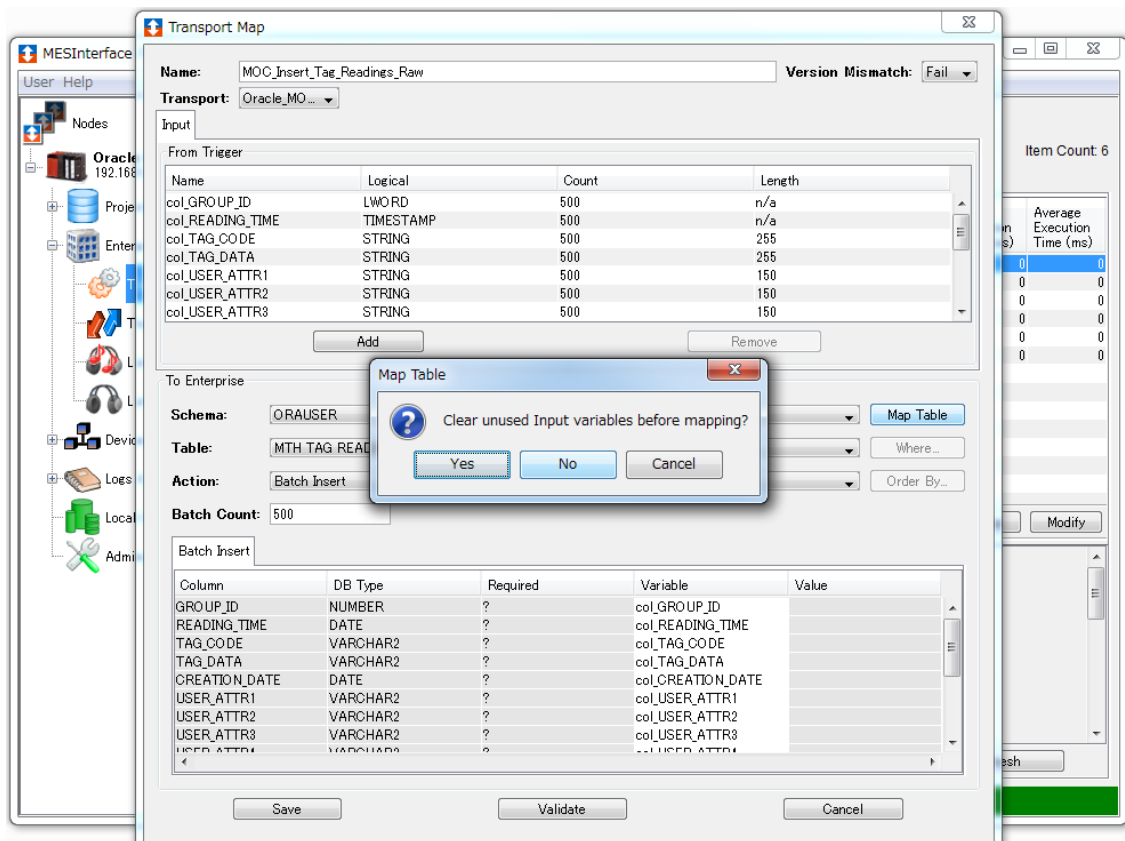
This will display the Transport Map edit window. Select your schema from the schema list.



Next, select the **MTH\_TAG\_READINGS\_RAW** interface table from the table list. The columns for this table will be listed at the bottom of the window.



Click the **Map Table** button, and the **No** button in the message pop up box.



The Input variables will be automatically mapped to their corresponding columns in the table. Click the Save button and the transport map configuration will be completed.

**Transport Map**

Name: MOC\_Insert\_Tag\_Readings\_Raw Version Mismatch: Fail

Transport: Oracle\_MO...

Input

From Trigger

| Name             | Logical | Count | Length |
|------------------|---------|-------|--------|
| col_GROUP_ID     | 0       | 500   | n/a    |
| col_READING_TIME | 0       | 500   | n/a    |
| col_TAG_CODE     | 0       | 500   | n/a    |
| col_TAG_DATA     | 0       | 500   | n/a    |
| col_USER_ATTR1   | 0       | 500   | n/a    |
| col_USER_ATTR2   | 0       | 500   | n/a    |
| col_USER_ATTR3   | 0       | 500   | n/a    |

Add Remove

To Enterprise

Schema: ORAUSER Map Table

Table: MTH TAG READINGS RAW Where...

Action: Batch Insert Order By...

Batch Count: 500

Batch Insert

| Column        | DB Type  | Required | Variable          | Value |
|---------------|----------|----------|-------------------|-------|
| GROUP_ID      | NUMBER   | ?        | col_GROUP_ID      |       |
| READING_TIME  | DATE     | ?        | col_READING_TIME  |       |
| TAG_CODE      | VARCHAR2 | ?        | col_TAG_CODE      |       |
| TAG_DATA      | VARCHAR2 | ?        | col_TAG_DATA      |       |
| CREATION_DATE | DATE     | ?        | col_CREATION_DATE |       |
| USER_ATTR1    | VARCHAR2 | ?        | col_USER_ATTR1    |       |
| USER_ATTR2    | VARCHAR2 | ?        | col_USER_ATTR2    |       |
| USER_ATTR3    | VARCHAR2 | ?        | col_USER_ATTR3    |       |
| USER_ATTR4    | VARCHAR2 | ?        | col_USER_ATTR4    |       |

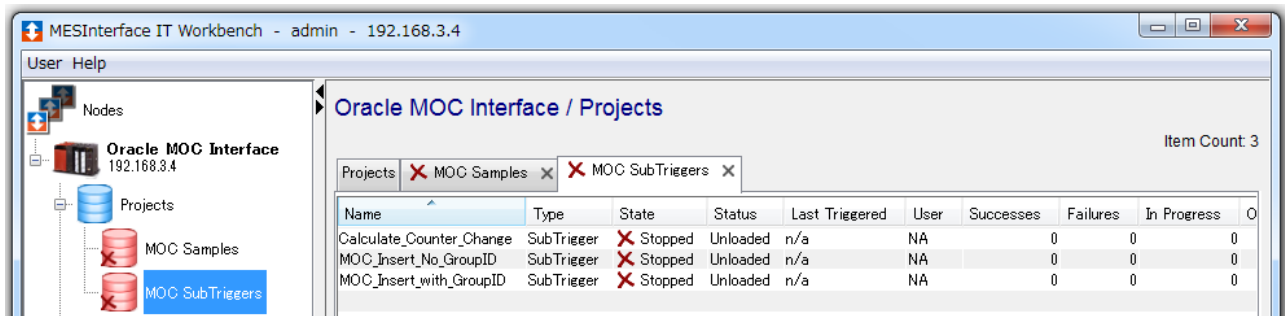
Save Validate Cancel

This completes the configuration of the Oracle MOC Extension. The next section will review how to customize the Oracle MOC Extension for your specific data collection requirements.

# Chapter 4: Customizing Transactions

## Overview of the Projects

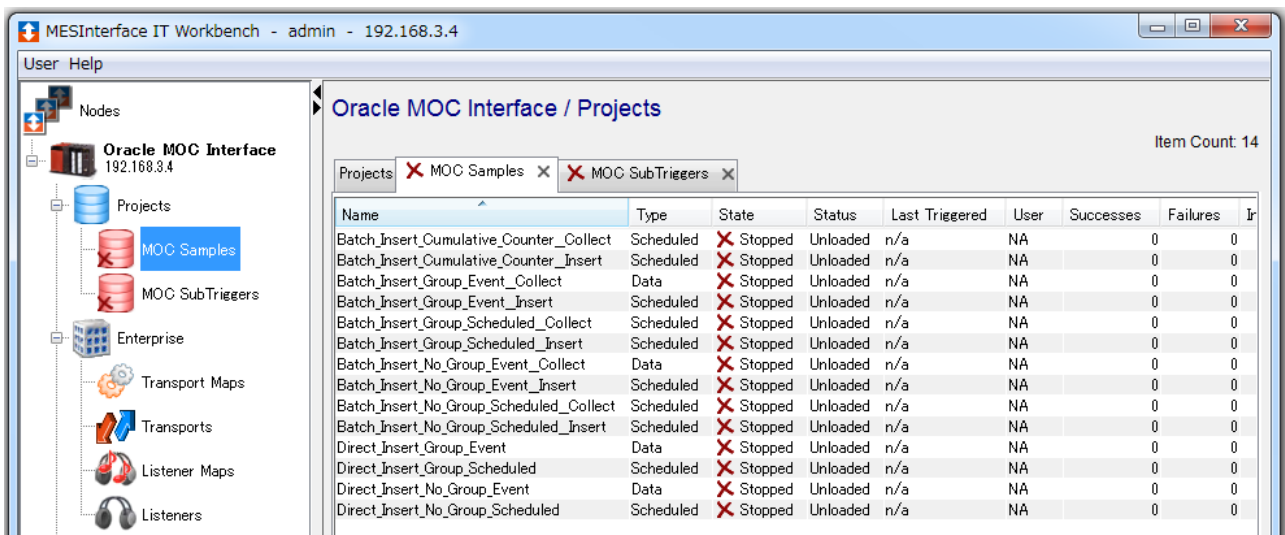
There are two projects included with the Oracle MOC Extension, MOC SubTriggers and MOC Samples. The MOC SubTriggers project contains two critical subtriggers that collect the records from temporary storage in the local database and inserts them into the Oracle Manufacturing Operations Center interface table. One of these subtriggers,



| Name                     | Type       | State   | Status   | Last Triggered | User | Successes | Failures | In Progress | Item Count |
|--------------------------|------------|---------|----------|----------------|------|-----------|----------|-------------|------------|
| Calculate_Counter_Change | SubTrigger | Stopped | Unloaded | n/a            | NA   | 0         | 0        | 0           | 0          |
| MOC_Insert_No_GroupID    | SubTrigger | Stopped | Unloaded | n/a            | NA   | 0         | 0        | 0           | 0          |
| MOC_Insert_with_GroupID  | SubTrigger | Stopped | Unloaded | n/a            | NA   | 0         | 0        | 0           | 0          |

MOC\_Insert\_with\_GroupID, uses the Oracle Group ID to organize the records, and the other, MOC\_Insert\_No\_GroupID, does not. The third subtrigger, Calculate\_Counter\_Change, is used calculate the difference between two readings of a device counter variable. It is assumed that this counter variable will be reset at a predetermined level, and this subtrigger can also calculate the difference when the reset occurs between readings. Other subtriggers that are specific to your Oracle Manufacturing Operations Center implementation can be added to this project, as well.

The MOC Samples project contains a variety of sample triggers that you can use as a guide to create the triggers for your specific data collection requirements. Among the samples are batch



| Name                                    | Type      | State   | Status   | Last Triggered | User | Successes | Failures | Item Count |
|---|-----------|---------|----------|----------------|------|-----------|----------|------------|
| Batch_Insert_Cumulative_Counter_Collect | Scheduled | Stopped | Unloaded | n/a            | NA   | 0         | 0        | 0          |
| Batch_Insert_Cumulative_Counter_Insert  | Scheduled | Stopped | Unloaded | n/a            | NA   | 0         | 0        | 0          |
| Batch_Insert_Group_Event_Collect        | Data      | Stopped | Unloaded | n/a            | NA   | 0         | 0        | 0          |
| Batch_Insert_Group_Event_Insert         | Scheduled | Stopped | Unloaded | n/a            | NA   | 0         | 0        | 0          |
| Batch_Insert_Group_Scheduled_Collect    | Scheduled | Stopped | Unloaded | n/a            | NA   | 0         | 0        | 0          |
| Batch_Insert_Group_Scheduled_Insert     | Scheduled | Stopped | Unloaded | n/a            | NA   | 0         | 0        | 0          |
| Batch_Insert_No_Group_Event_Collect     | Data      | Stopped | Unloaded | n/a            | NA   | 0         | 0        | 0          |
| Batch_Insert_No_Group_Event_Insert      | Scheduled | Stopped | Unloaded | n/a            | NA   | 0         | 0        | 0          |
| Batch_Insert_No_Group_Scheduled_Collect | Scheduled | Stopped | Unloaded | n/a            | NA   | 0         | 0        | 0          |
| Batch_Insert_No_Group_Scheduled_Insert  | Scheduled | Stopped | Unloaded | n/a            | NA   | 0         | 0        | 0          |
| Direct_Insert_Group_Event               | Data      | Stopped | Unloaded | n/a            | NA   | 0         | 0        | 0          |
| Direct_Insert_Group_Scheduled           | Scheduled | Stopped | Unloaded | n/a            | NA   | 0         | 0        | 0          |
| Direct_Insert_No_Group_Event            | Data      | Stopped | Unloaded | n/a            | NA   | 0         | 0        | 0          |
| Direct_Insert_No_Group_Scheduled        | Scheduled | Stopped | Unloaded | n/a            | NA   | 0         | 0        | 0          |

triggers, which collect records from multiple transactions and insert them into Oracle as a single batch insert, and row transactions, which collect records from a single transaction and inserts them into Oracle. The batch triggers are comprised of a pair of triggers. One trigger to collect the data from each transaction and place it in local storage and the other to pull the data from local storage and batch insert it into Oracle.

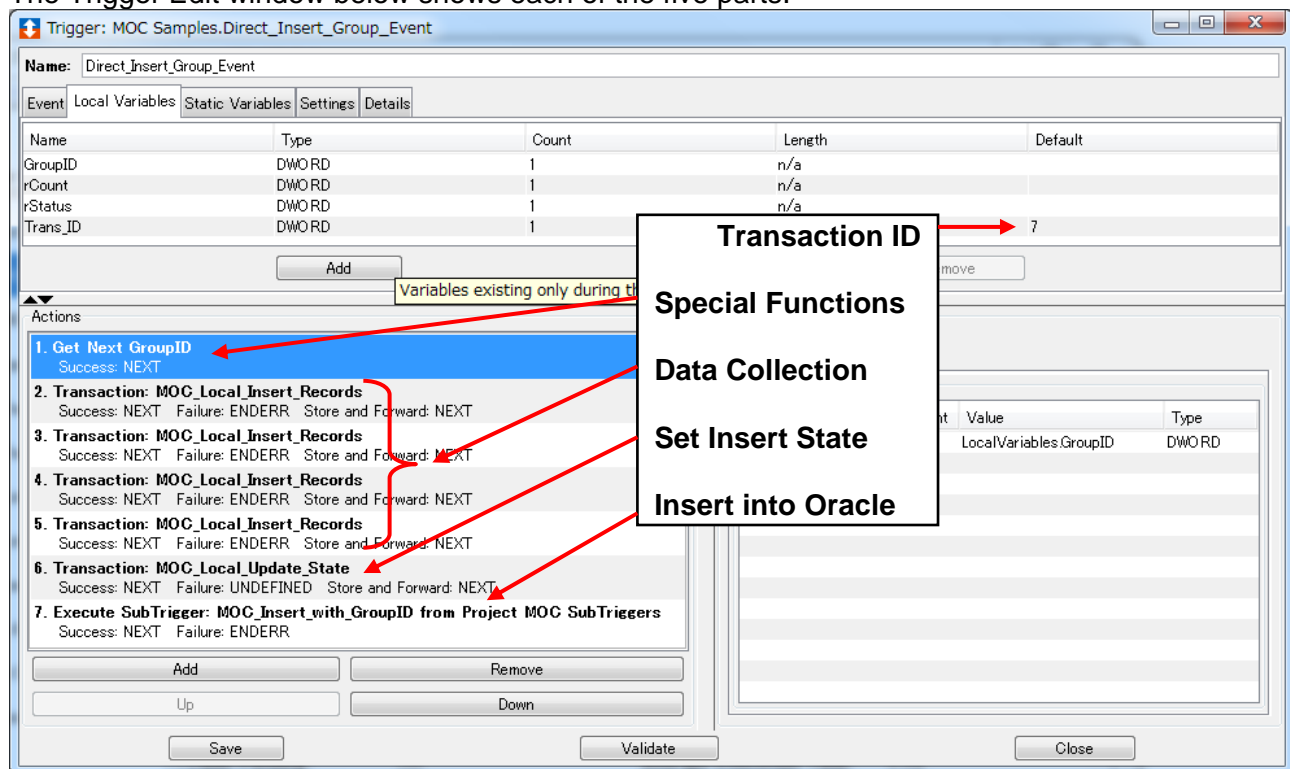
There are also transaction samples which are triggered by events, which are triggered periodically, which use the Oracle Group ID, and which use cumulative counter variables. Each of these conditions will be discussed through setting up a sample transaction.

## Oracle MOC Extension Trigger Structure

There are five parts to the Oracle MOC Extension trigger structure:

1. Transaction ID
2. Special Functions
3. Data Collection
4. Set Insert State
5. Insert into Oracle

The Trigger Edit window below shows each of the five parts.



## Transaction ID

The transaction ID is an internal identifier which keeps all the records from each trigger grouped together for processing within the MESInterface IT node. The transaction ID must be unique for each direct trigger or batch trigger pair which sends data records to Oracle.

## Special Functions

Special functions are any data manipulation or identifier actions which are required for the Oracle records. These are entered as the first actions in preparation for data collection. Examples of special functions are the Oracle Group ID and counter value calculations.

## Data Collection

The data collection actions will follow any special functions. There will be one action for each data element that is being collected in this trigger. In this example there are four data elements. All the information that is required by Oracle Manufacturing Operations Center for the data element is entered with each action. The details of the data collection action will be covered in the *Setting up a Direct Insert Transaction* section.

## Set Insert State

The set insert state calls a transaction which sets a state variable in the collected records, signifying all the data for this trigger has been successfully collected and is ready to insert into Oracle. It is required directly after the data collection actions in the action list.

## Insert into Oracle

The final part is the execution of a subtrigger that inserts the data into Oracle. For a direct insert, this part will be the last item in the action list. For batch inserts, it will be the only action in a separate trigger. Batch insert structure will be illustrated in the *Setting up a Batch Insert Transaction* section.

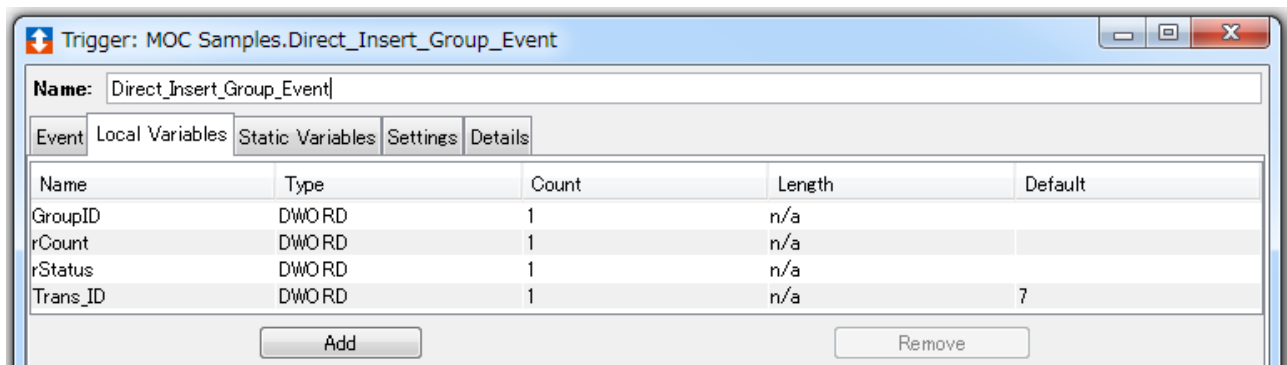
## Setting up a Direct Insert Transaction

The first sample transaction we will customize use's the Direct\_Insert\_Group\_Event trigger as a starting point. This is a data event trigger that fires when a device variable changes value. More information on configuring the trigger event type can be found in the *MESInterface IT User's Guide*. The Direct\_Insert\_Group\_Event trigger collects multiple data variables and inserts the records into the Oracle Manufacturing Operations Center interface table as a single transaction. It uses the Oracle Group ID to group the values together for processing.

To customize this trigger for your data collection requirements, there are three parts that need to be modified: The transaction ID, the Group ID, and device data information. To begin, click the **MOC Samples** icon under **Projects**. Then select the Direct\_Insert\_Group\_Event trigger from the MOC Samples trigger list and click **Edit**. Each of the three parts can be modified through the Trigger Edit window.

## Setting the Transaction ID

To set the transaction ID, select the **Local Variables** tab near the top of the Trigger Edit window and add a default value to the **Trans\_ID** variable. This can be any four-byte integer value you want, as long as each transaction ID is unique for this instance of MESInterface IT. In the example following, the transaction ID is 7.



Trigger: MOC Samples.Direct\_Insert\_Group\_Event

Name: Direct\_Insert\_Group\_Event

Event Local Variables Static Variables Settings Details

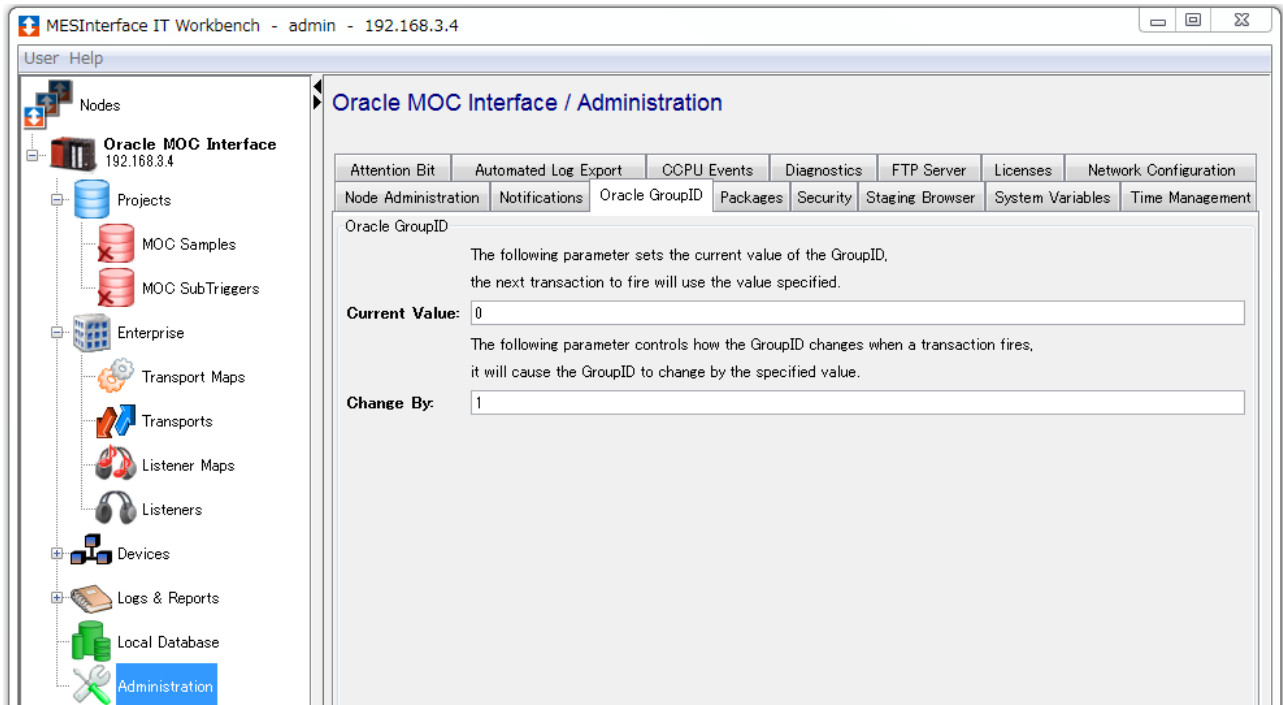
| Name     | Type  | Count | Length | Default |
|----------|-------|-------|--------|---------|
| GroupID  | DWORD | 1     | n/a    |         |
| rCount   | DWORD | 1     | n/a    |         |
| rStatus  | DWORD | 1     | n/a    |         |
| Trans_ID | DWORD | 1     | n/a    | 7       |

Add Remove



## Setting the Group ID

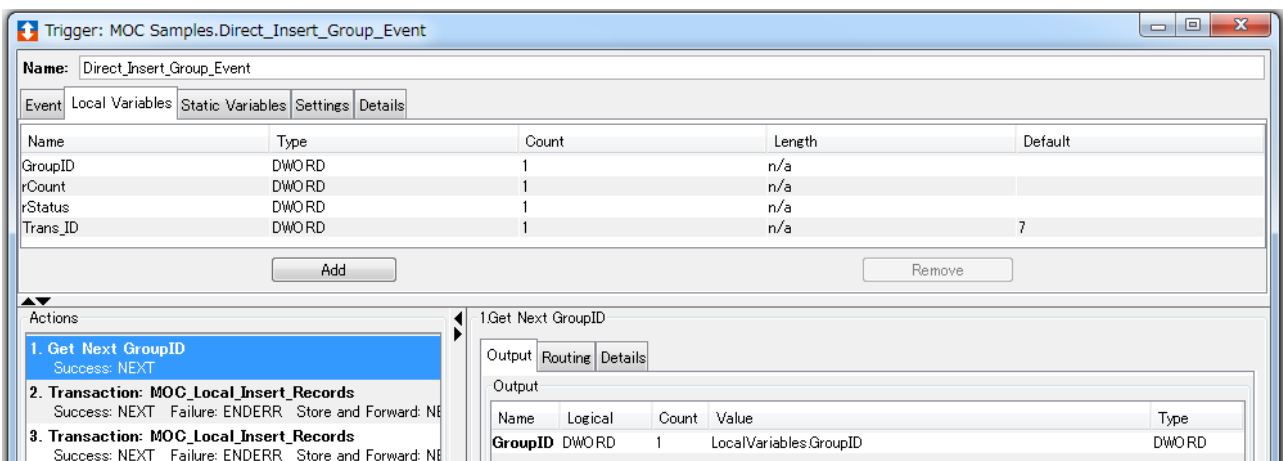
The Group ID is an identifier used by Oracle Manufacturing Operations Center to group records collected in the same transaction for processing. To set the Group ID parameters, Click on the **Administration** icon and select the **Oracle GroupID** tab.



There are two parameters to set, the Current Value and Change By. The Current Value is the value that the next Group ID will receive and the Change By is the increment between Group ID's. In the example below, the starting Group ID will be 1, and it will increment by 1 for each Group ID, as in 1, 2, 3, 4 and so on.

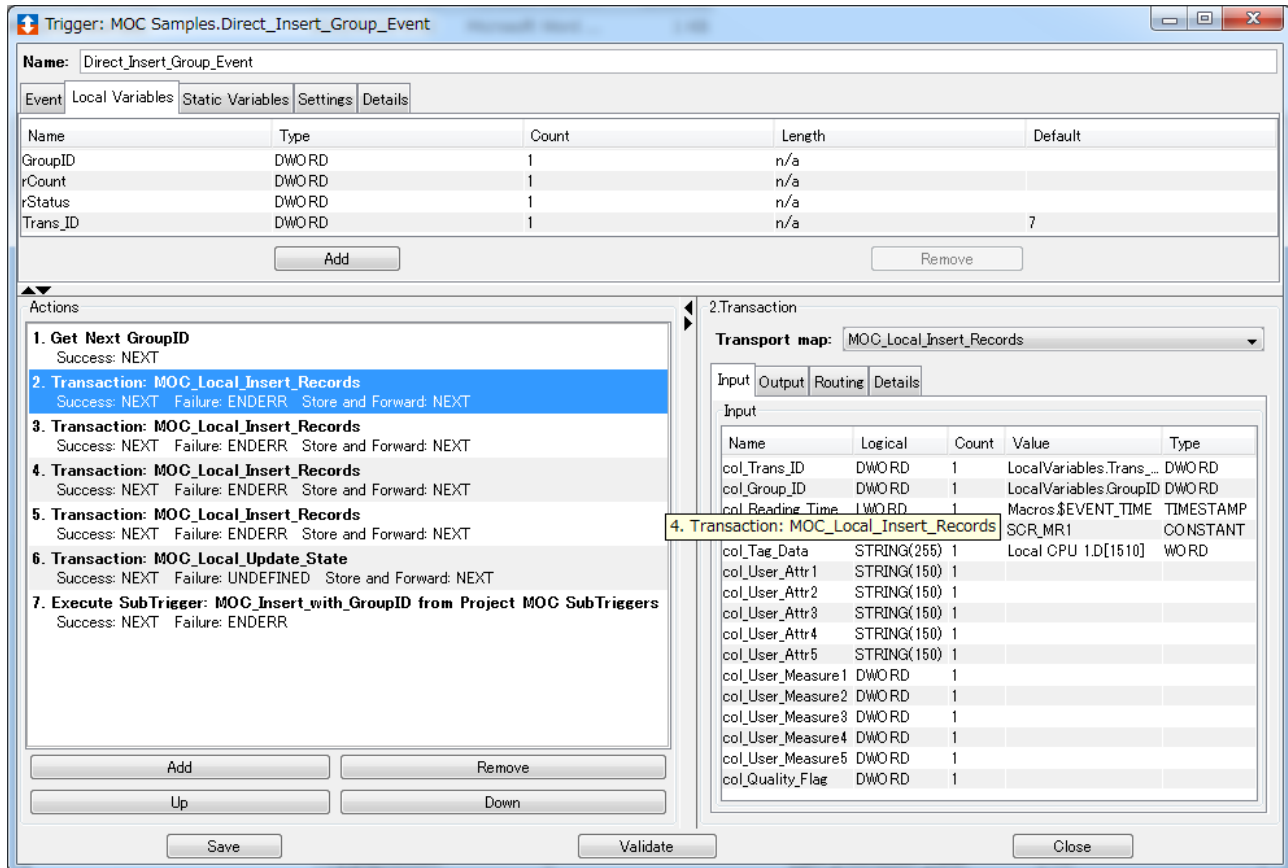
You should only have to set these parameters once, during the implementation of the Oracle MOC Extension.

To use the Group ID, you simply add the **Get Next GroupID** action to your action list. In the example below, the next Group ID is assigned to the local variable GroupID for later use.



## Adding Data Collection Actions

To add a data collection action, either add an enterprise transaction using the transport map, MOC\_Local\_Insert\_Records or modify an existing action from the samples. In the example below there are four data collection actions. The first data collection record is highlighted, and its parameters are displayed on the lower right section of the Trigger Edit window.



For this record there are five fields that are used for data collection:

1. col\_Trans\_ID, the transaction ID
2. col\_Group\_ID, the Oracle GroupID
3. col\_Reading\_Time, the time the data was read
4. col\_Tag\_Code, the name of the data tag
5. col\_Tag\_Data, the value of the data tag.

These are the five key fields for data collection when an Oracle Group ID is used. The col\_Trans\_ID should be set to the local variable Trans\_ID. The col\_Group\_ID should be set to the local variable GroupID. If the Oracle Group ID is not used, then the col\_Group\_ID would be left blank. The col\_Reading\_Time should be set to the current timestamp by using the \$EVENT\_TIME macro. The col\_Tag\_Code should be set to the tag code configured in the Oracle Manufacturing Operations Center for this data tag. The last field used, col\_Tag\_Data, is the address of the data tag in the device. The next ten fields consist of user attribute and user measurement fields and can be used based on your configuration of the Oracle Manufacturing Operations Center. The last field, col\_Quality\_Flag, is not currently supported by the MESInterface IT node. Once these fields are configured, and the trigger is started, device data can be directly inserted into the Oracle Manufacturing Operations Center using MESInterface IT.

## Setting up a Batch Insert Transaction

The second transaction we will customize use's the Batch\_Insert\_Cumulative\_Counter\_\_Collect trigger as a starting point. This is a scheduled event trigger that fires every 15 seconds and collects multiple data variables and inserts the records into the Oracle Manufacturing Operations Center interface table as a batch insert. It also uses the Oracle Group ID to group the values together for processing, and has two counter variables that need the change in counter value to be inserted into Oracle. The collect trigger stores the data records in the local database for each time it executes.

**Trigger: MOC Samples.Batch\_Insert\_Cumulative\_Counter\_\_Collect**

Name: Batch\_Insert\_Cumulative\_Counter\_\_Collect

Event Local Variables Static Variables Settings Details

| Name          | Type  | Count | Length | Default |
|---------------|-------|-------|--------|---------|
| GroupID       | DWORD | 1     | n/a    |         |
| OUT_MR1_Value | DWORD | 1     | n/a    |         |
| rCount        | DWORD | 1     | n/a    |         |
| rStatus       | DWORD | 1     | n/a    |         |
| SCR_MR1_Value | DWORD | 1     | n/a    |         |
| TransID       | DWORD | 1     | n/a    | 9       |

Add Remove

Actions

- 1. Get Next GroupID  
Success: NEXT
- 7. Execute SubTrigger: Calculate\_Counter\_Change from Project MOC SubTriggers  
Success: NEXT Failure: ENDERR
- 8. Execute SubTrigger: Calculate\_Counter\_Change from Project MOC SubTriggers  
Success: NEXT Failure: ENDERR
- 2. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT
- 3. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT
- 4. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT
- 5. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT
- 6. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT
- 9. Transaction: MOC\_Local\_Update\_State  
Success: NEXT Failure: ENDERR Store and Forward: NEXT

Add Remove Up Down Save Validate Close

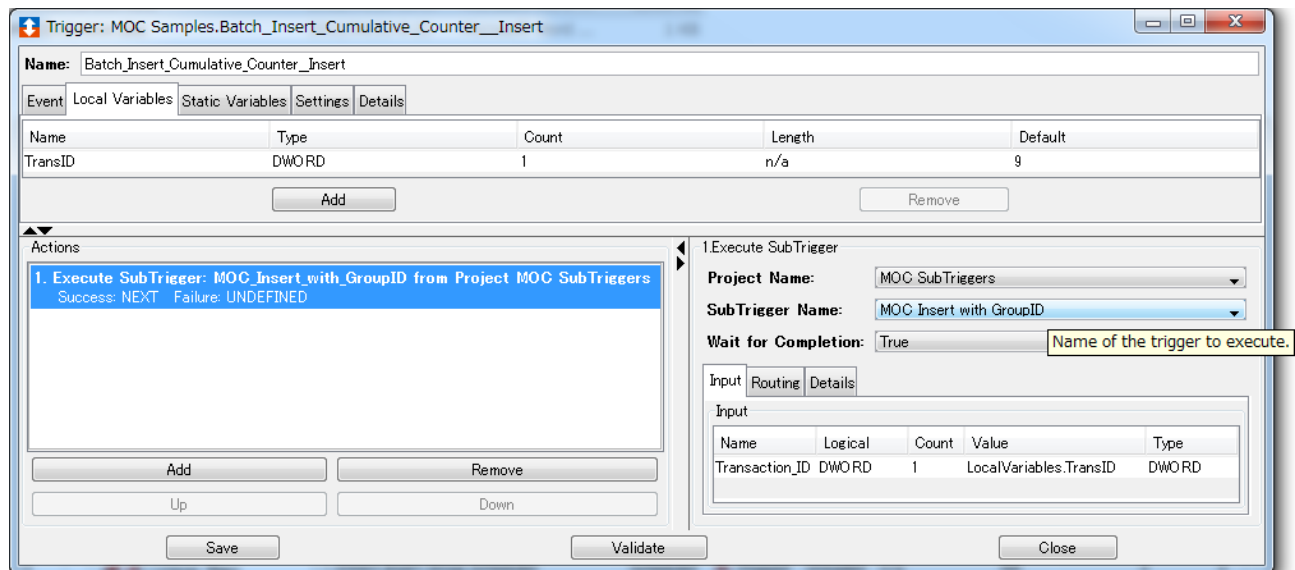
1.Get Next GroupID

Output Routing Details

Output

| Name    | Logical | Count | Value                  | Type  |
|---------|---------|-------|------------------------|-------|
| GroupID | DWORD   | 1     | LocalVariables.GroupID | DWORD |

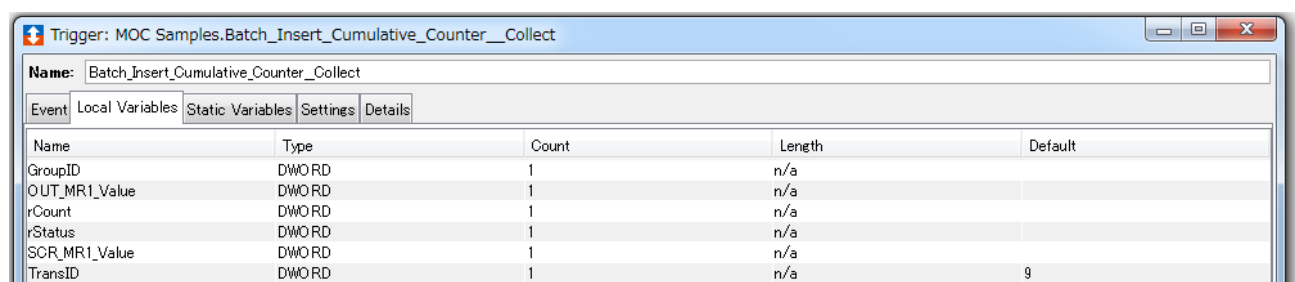
The Batch\_Insert\_Cumulative\_Counter\_\_Insert trigger selects the records from the local database, based on the transaction ID, and inserts them into Oracle every two minutes. Using a batch insert methodology helps to reduce the network traffic and database activity.



To customize these triggers for your data collection requirements, there are four parts that need to be modified: The transaction ID, the Group ID, the cumulative counter values and device data information. To begin, click the **MOC Samples** icon under **Projects**. Then select the Batch\_Insert\_Cumulative\_Counter\_\_Collect trigger from the MOC Samples trigger list and click **Edit**. Each of the parts can be modified through the Trigger Edit window.

## Setting the Transaction ID

To set the transaction ID for both triggers, select the **Local Variables** tab near the top of the Trigger Edit window and add a default value to the **Trans\_ID** variable. This can be any four-byte integer value you want, as long as each transaction ID is unique for this instance of MESInterface IT. In the example following, the transaction ID is 9 for both triggers.



## Setting the Group ID

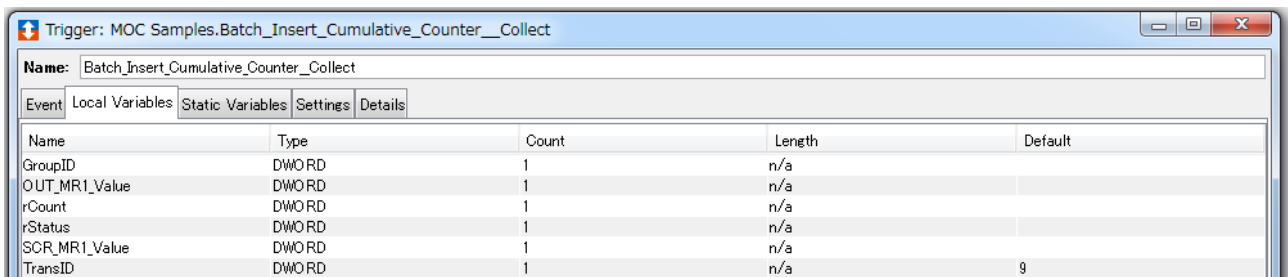
To use the Group ID, you simply add the **Get Next GroupID** action to your action list. In the example below, the next Group ID is assigned to the local variable GroupID for later use.

## Using a Counter Variable

The counter variable assumes that values from a counter in a device are being collected, and the difference in the counter values between two readings is the data required by the Oracle Manufacturing Operations Center. This difference is the actual count over a given period of time. For example, if the counter value is 5 at the first reading and 15 at the second reading, then the difference or count would be 10.

A counter also can reset after it reaches a defined value. For instance, a counter can start at 0 and count to 999, then reset to 0 and continue counting. To accurately calculate the difference or count value, the formula must also calculate the difference when a reset has occurred between readings. All of these considerations were used in developing the Calculate\_Counter\_Change subtrigger.

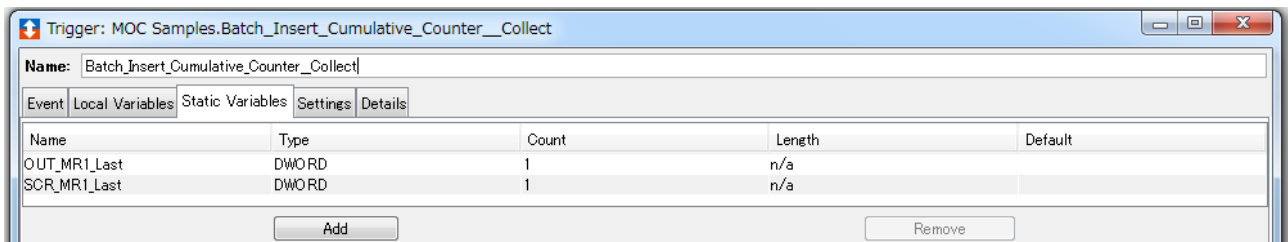
To use the Calculate\_Counter\_Change subtrigger, you need to define a local variable and a static variable. The local variable will be set to the calculated count, and the static variable will hold the last value of the counter for calculating the next count. In the example below, the local variable OUT\_MR1\_Value will be set to the calculated count for insert into Oracle.



The screenshot shows a window titled "Trigger: MOC Samples.Batch\_Insert\_Cumulative\_Counter\_Collect". The "Name" field contains "Batch\_Insert\_Cumulative\_Counter\_Collect". The "Local Variables" tab is selected, displaying a table with the following data:

| Name          | Type  | Count | Length | Default |
|---------------|-------|-------|--------|---------|
| GroupID       | DWORD | 1     | n/a    |         |
| OUT_MR1_Value | DWORD | 1     | n/a    |         |
| rCount        | DWORD | 1     | n/a    |         |
| rStatus       | DWORD | 1     | n/a    |         |
| SCR_MR1_Value | DWORD | 1     | n/a    |         |
| TransID       | DWORD | 1     | n/a    | 9       |

There is also a static variable, OUT\_MR1\_Last, which will hold the last value collected from a counter tag.



The screenshot shows the same window as above, but with the "Static Variables" tab selected. It displays a table with the following data:

| Name         | Type  | Count | Length | Default |
|--------------|-------|-------|--------|---------|
| OUT_MR1_Last | DWORD | 1     | n/a    |         |
| SCR_MR1_Last | DWORD | 1     | n/a    |         |

Below the table are "Add" and "Remove" buttons.

When the Calculate\_Counter\_Change subtrigger is added to the action list, there are four input parameters required and two out parameters. The input parameters are as follows:

1. Current – the current counter value retrieved directly from the device.
2. Last – the last counter value retrieved from the static variable holding the last value.
3. Reset – the highest value the counter will reach before resetting.
4. Start – the starting value after a counter reset.

These input values are highlighted in the red circle on the following trigger edit window.

**Trigger: MOC Samples.Batch\_Insert\_Cumulative\_Counter\_\_Collect**

Name: Batch\_Insert\_Cumulative\_Counter\_\_Collect

Event Local Variables Static Variables Settings Details

| Name          | Type  | Count | Length | Default |
|---------------|-------|-------|--------|---------|
| GroupID       | DWORD | 1     | n/a    |         |
| OUT_MR1_Value | DWORD | 1     | n/a    |         |
| rCount        | DWORD | 1     | n/a    |         |
| rStatus       | DWORD | 1     | n/a    |         |
| SCR_MR1_Value | DWORD | 1     | n/a    |         |
| TransID       | DWORD | 1     | n/a    | 9       |

7 Execute SubTrigger

Project Name: MOC SubTriggers

SubTrigger Name: Calculate\_Counter\_Change

Wait for Completion: True

Input Output Routing Details

| Name    | Logical | Count | Value                        | Type     |
|---------|---------|-------|------------------------------|----------|
| Current | DWORD   | 1     | Local CPU 1.D[1506]          | WORD     |
| Last    | DWORD   | 1     | StaticVariables.OUT_MR1_L... | DWORD    |
| Reset   | DWORD   | 1     | 999                          | CONSTANT |
| Start   | DWORD   | 1     | 0                            | CONSTANT |

1. Get Next GroupID  
Success: NEXT

7. Execute SubTrigger: Calculate\_Counter\_Change from Project MOC SubTriggers  
Success: NEXT Failure: ENDERR

8. Execute SubTrigger: Calculate\_Counter\_Change from Project MOC SubTriggers  
Success: NEXT Failure: ENDERR

2. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT

3. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT

4. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT

5. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT

6. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT

9. Transaction: MOC\_Local\_Update\_State  
Success: NEXT Failure: ENDERR Store and Forward: NEXT

Add Remove Up Down Save Validate Close

The output parameters are as follows:

1. Delta – the change in the two counter values and is assigned to the local variable holding the count.
2. NewLast – the current value from the device becomes the new last value and is assigned to the static variable holding the last value.

These output values are highlighted in the red circle on the following trigger edit window.

**Trigger: MOC Samples.Batch\_Insert\_Cumulative\_Counter\_\_Collect**

Name: Batch\_Insert\_Cumulative\_Counter\_\_Collect

Event Local Variables Static Variables Settings Details

| Name          | Type  | Count | Length | Default |
|---------------|-------|-------|--------|---------|
| GroupID       | DWORD | 1     | n/a    |         |
| OUT_MR1_Value | DWORD | 1     | n/a    |         |
| rCount        | DWORD | 1     | n/a    |         |
| rStatus       | DWORD | 1     | n/a    |         |
| SCR_MR1_Value | DWORD | 1     | n/a    |         |
| TransID       | DWORD | 1     | n/a    | 9       |

Add Remove

Actions

1. Get Next GroupID  
Success: NEXT
7. Execute SubTrigger: Calculate\_Counter\_Change from Project MOC SubTriggers  
Success: NEXT Failure: ENDERR
8. Execute SubTrigger: Calculate\_Counter\_Change from Project MOC SubTriggers  
Success: NEXT Failure: ENDERR
2. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT
3. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT
4. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT
5. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT
6. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT
9. Transaction: MOC\_Local\_Update\_State  
Success: NEXT Failure: ENDERR Store and Forward: NEXT

Add Remove Up Down Save Validate Close

7. Execute SubTrigger

Project Name: MOC SubTriggers

SubTrigger Name: Calculate\_Counter\_Change

Wait for Completion: True

Input Output Routing Details

Output

| Name    | Logical | Count | Value                        | Type  |
|---------|---------|-------|------------------------------|-------|
| Delta   | DWORD   | 1     | LocalVariables.OUT_MR1_Value | DWORD |
| NewLast | DWORD   | 1     | StaticVariables.OUT_MR1_Last | DWORD |

## Adding Data Collection Actions

Adding data collection actions is the same for both direct and batch insert triggers. The minor difference in this example is for the cumulative counter values. Instead of assigning the device tag value to be inserted into Oracle, the local variable that holds the difference in counter values is assigned. This is the Delta output value from the Calculate\_Counter\_Change subtrigger. This assignment is highlighted in the circle on the following trigger edit window.

Trigger: MOC Samples.Batch\_Insert\_Cumulative\_Counter\_\_Collect

Name: Batch\_Insert\_Cumulative\_Counter\_\_Collect

Event Local Variables Static Variables Settings Details

| Name          | Type  | Count | Length | Default |
|---------------|-------|-------|--------|---------|
| GroupID       | DWORD | 1     | n/a    |         |
| OUT_MR1_Value | DWORD | 1     | n/a    |         |
| rCount        | DWORD | 1     | n/a    |         |
| rStatus       | DWORD | 1     | n/a    |         |
| SCR_MR1_Value | DWORD | 1     | n/a    |         |
| TransID       | DWORD | 1     | n/a    | 9       |

Add Remove

Actions

1. Get Next GroupID  
Success: NEXT
7. Execute SubTrigger: Calculate\_Counter\_Change from Project MOC SubTriggers  
Success: NEXT Failure: ENDERR
8. Execute SubTrigger: Calculate\_Counter\_Change from Project MOC SubTriggers  
Success: NEXT Failure: ENDERR
2. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT
3. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT
4. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT
5. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT
6. Transaction: MOC\_Local\_Insert\_Records  
Success: NEXT Failure: ENDERR Store and Forward: NEXT
9. Transaction: MOC\_Local\_Update\_State  
Success: NEXT Failure: ENDERR Store and Forward: NEXT

Add Remove Up Down Save Validate Close

2.Transaction

Transport map: MOC\_Local\_Insert\_Records

Input Output Routing Details

Input

| Name              | Logical     | Count | Value                        | Type      |
|-------------------|-------------|-------|------------------------------|-----------|
| col_Trans_ID      | DWORD       | 1     | LocalVariables.TransID       | DWORD     |
| col_Group_ID      | DWORD       | 1     | LocalVariables.GroupID       | DWORD     |
| col_Reading_Time  | LWORD       | 1     | Macros\$EVENT_TIME           | TIMESTAMP |
| col_Tag_Code      | STRING(255) | 1     | OUT_MR1                      | CONSTANT  |
| col_Tag_Data      | STRING(255) | 1     | LocalVariables.OUT_MR1_Value | DWORD     |
| col_User_Attr1    | STRING(150) | 1     |                              |           |
| col_User_Attr2    | STRING(255) | 1     |                              |           |
| col_User_Attr3    | STRING(150) | 1     |                              |           |
| col_User_Attr4    | STRING(150) | 1     |                              |           |
| col_User_Attr5    | STRING(150) | 1     |                              |           |
| col_User_Measure1 | DWORD       | 1     |                              |           |
| col_User_Measure2 | DWORD       | 1     |                              |           |
| col_User_Measure3 | DWORD       | 1     |                              |           |
| col_User_Measure4 | DWORD       | 1     |                              |           |
| col_User_Measure5 | DWORD       | 1     |                              |           |
| col_Quality_Flag  | DWORD       | 1     |                              |           |

Once these fields are configured, and both the collection and insert triggers are started, device data can be batch inserted into the Oracle Manufacturing Operations Center using MESInterface IT.

## Customizing the Subtriggers

The subtriggers that are provided with the Oracle MOC Extension should not require customization to perform the functions they were designed for. The only exception is if you require using the user attribute or user measurement columns from the Oracle Manufacturing Operations Center interface table. If any of these columns are used, you should create a new subtrigger by duplicating an existing one, and include the new columns in the insert transaction. You will also need to create a new transport map by duplicating one of the existing MOC\_Local\_Select\_Records\_#### transport maps to select the new user attribute or measurement columns from the local database.

Other subtriggers can be added to provide special functions, like Calculate\_Counter\_Change.

This concludes the *Customizing Transactions* section.



# Chapter 5: Troubleshooting

The following provides troubleshooting help for problems specific to the Oracle MOC Extension. Additional help can be found in the *MESInterface IT Troubleshooting Guide*.

**The package file or export files fail during import.**

Contact ILS Technology support for replacement files.

**The Oracle\_MOC transport does not validate.**

Check your database name, host address, port number, user name and password. If all of these are correct, check your communication path to your Oracle database, and make sure the Oracle database is operating properly.

**I can't find the MTH\_TAG\_READINGS\_RAW table in the MOC\_Insert\_Tag\_Readings\_Raw transport map table list.**

Select the correct schema from the Schema list, then locate the MTH\_TAG\_READINGS\_RAW table. If you cannot find the correct schema, check with your database administrator to make sure your user name has access to the required schema.

**I've started a customized sample trigger, but it won't run.**

Check to make sure the Project it's in has been started.

**I've started a customized sample trigger, but it fails immediately.**

Check to make sure the MOC SubTriggers project has been started, and that all the subtriggers in that project have been started.

**I've started a customized sample trigger, but the transactions are going into the store and forward queue.**

Validate the Oracle\_MOC transport, and make sure the communication path to your Oracle database and the Oracle database are operating properly.

**I'm using a batch insert transaction, but the data is staying in the local database.**

Check to make sure the corresponding insert trigger has been started. If it has, check to make sure that the transaction ID for both the collect and insert triggers are the same.

**The Oracle Group ID is not being inserted into the Oracle Manufacturing Operations Center interface table with my data.**

Check to make sure that you are using the Get Next GroupID special function, that you are populating the col\_Group\_ID in your data collection actions, and that you are using the MOC\_Insert\_with\_GroupID subtrigger to insert the records into Oracle.

## Index

### A

Administration icon, 4

### B

batch triggers, 19

### C

Configure the Oracle Transport, 13

Configure the Transport Map, 15

### G

Group ID, 21

### I

Installing the Project, 9

### L

local database table, 7

### O

Oracle Manufacturing Operation Center, 3

Oracle MOC Extension, 3

Oracle MOC Package, 4

### P

Packages tab., 4

projects, 18

### S

schema, 15

subtriggers, 18, 28

### T

triggers, 18

**"deviceWISE™ enabled"**

deviceWISE is a trademark of ILS Technology LLC.

Microsoft, Windows, Windows Vista, SQL server, and Visual Studio are registered trademarks of Microsoft Corporation in the United States and other countries.

Sun, Sun Microsystems, Java, J2ME and J2SE are either trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Oracle is a registered trademark of Oracle Corporation.

Adobe and Acrobat are registered trademarks of Adobe Systems Incorporation.

Pentium and Celeron are trademarks of Intel Corporation in the united States and other countries.

Ethernet is a trademark of Xerox Co., Ltd. in the United States.

CompactFlash is a trademark of SanDisk Corporation.

Other company names used in this document are trademarks or registered trademarks of respective owners.

The Program may contain some or all of the following third party components:

EZSocket Copyright (C) 2008 Mitsubishi Electric Corporation

iconv Copyright (C) 2007 Free Software Foundation

sqlite Copyright (C) 2008 Christian Werner

libxml2 Copyright (C) 2008 xmlsoft.org

log4j as originally developed by Apache.org

OpenSSL Toolkit, Open SSL Project, OpenSSL Copyright (C) 1998-2008 the Open SSL Project;

This product includes cryptographic software written by

Eric Young (eay@cryptsoft.com) and by Tim Hudson (tjh@cryptsoft.com)

Jamaica JVM Copyright (C) 2008 Aicas

# MESInterface IT

## Oracle MOC Extension Installation and User's Guide

|                |  |
|----------------|--|
| MODEL          |  |
| MODEL<br>CODE  |  |
| BAD-804Q015-A1 |  |



HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN  
NAGOYA WORKS : 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA, JAPAN

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.